



TUGAS AKHIR - TE 145561

Aplikasi Kontrol *Mixer* Audio Berbasis *Android*

Dyah Ayuningtyas
NRP 2214030044
Nafiqul Ihsan
NRP 2214030111

Dosen Pembimbing
Eko Pramunanto, ST., MT.
NIP. 19661203 199412 1 001

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 145561

***Audio Mixer Control Application Based on
Android***

Dyah Ayuningtyas
NRP 2214030044
Nafiqul Ihsan
NRP 2214030111

Supervisor
Eko Pramunanto, ST., MT.
NIP. 19661203 199412 1 001

COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2017

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul

“Aplikasi Kontrol Mixer Audio Berbasis Android”

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.



Dyah Ayuningtyas
2214030044

Surabaya, 20 Juli 2017



Nafiqul Ihsan
2214030111

-----Halaman ini sengaja dikosongkan-----

APLIKASI KONTROL MIXER AUDIO BERBASIS ANDROID

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya

Pada

Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing



**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

Aplikasi Kontrol *Mixer* Audio Berbasis *Android*

Nama : Dyah Ayuningtyas
Nama : Nafiqul Ihsan
Pembimbing : Eko Pramunanto, ST., MT.

ABSTRAK

Baru ini perkembangan teknologi begitu maju terutama dalam hal *sound engineer*. Masih sering ditemui seorang *sound engineer*, secara manual mengatur potensio geser pada *mixer* audio, namun dalam monitoring keluaran yang dihasilkan *mixer* telah banyak menggunakan monitor digital. Pengaturan *mixer* audio yang dilakukan secara manual artinya *sound engineer* harus melakukan pengaturan *mixer* audio tersebut secara *stuck* yaitu mengatur *mixer* audio dimana *mixer* tersebut ditata dan harus memutar tombol secara nyata (*real*) tanpa dapat berpindah-pindah tempat untuk mendengarkan dan menyesuaikan secara langsung hasil pengaturan *mixer* yang diatur. Hal tersebut dirasa kurang praktis dan kurang memudahkan kinerja dari *sound engineer* tersebut. Oleh sebab itu, diperlukan suatu inovasi baru dalam pengaturan *mixer* audio secara praktis dengan menggunakan *Android* dengan *via Bluetooth*.

Pada aplikasi kontrol *mixer* audio berbasis *android* diciptakan *hardware* dan *software* yang masing-masing tersebut yaitu modul analog *Pre-Amp*, *Power Supply*, dan modul kontrol untuk *hardware*. Sedangkan *software* yang digunakan adalah *software* Arduino dan MIT App Inventor 2.

Pada proses pengaturan *mixer* audio tersebut, *mixer* audio memiliki 4 *channel* mendapat perintah penambahan dan pengurangan (*Volume*, *trebble*, *Bass*, dan *Mute*) dari Arduino yang dapat diatur melalui *Android* yang dikomunikasikan melalui *Bluetooth*. *Bluetooth* dapat memberi perintah dari jarak 25,5 meter dengan tingkatan suara antara -20dB sampai -59dB. Sehingga, memudahkan *sound engineer* dalam proses pengaturan *mixer* audio dari berbagai sudut ruang.

Kata Kunci : *Mixer* audio, *tone control* digital, *android*, *Bluetooth*.

-----Halaman ini sengaja dikosongkan-----

Audio Mixer Control Application Based on Android

Name : Dyah Ayuningtyas
Name : Nafiqul Ihsan
Supervisor : Eko Pramunanto, ST., MT.

ABSTRACT

These new technological developments are so advanced especially in terms of sound engineer. Still often encountered a sound engineer, manually set the slide potential on the audio mixer, but in monitoring the output of the mixer has been using a lot of digital monitors. Manually mixed audio mixer settings means that the sound engineer must adjust the audio mixer in a stuck setting the audio mixer in which the mixer is laid out and must rotate real buttons without moving to listen and adjust the mixer settings directly arranged. It is considered less practical and less easy performance of the sound engineer. Therefore, a new innovation is needed in setting the audio mixer practically by using Android via Bluetooth.

In the Application of audio-based mixer android mixer is created hardware and software each of which is analogue module Pre-Amp, power supply, and control module for hardware. While the software used is software Arduino and MIT App Inventor 2 which is combined in such a way that it can be run.

In the process of setting up the audio mixer, the audio mixer have 4 channels gets the addition and subtraction (Volume, trebble, Bass, and Mute) commands from Arduino which can be set via Android communicated via Bluetooth. Thus Bluetooth can give orders from a distance of 25.5 meters with sound level between -20 dB until -59dB . Thus, easy sound engineer in the process of setting the audio mixer from various angles of space.

Keyword : Mixer audio, tone control digital, android, Bluetooth.

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan diploma pada Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

Aplikasi Kontrol *Mixer* Audio Berbasis *Android*

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Eko Pramunanto, ST., MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini, kedua orang tua yang selalu memberikan doa, semangat, dan dukungannya kepada penulis. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 20 Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN

PERNYATAAN KEASLIAN TUGAS AKHIR	v
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Permasalahan	1
1.3 Batasan Masalah	2
1.4 Tujuan.....	2
1.5 Metodologi Penelitian.....	2
1.6 Sistematika Laporan	3
1.7 Relevansi	4
BAB II TEORI DASAR	5
2.1 Tinjauan Pustaka	5
2.2 IC TL081CN	6
2.3 MAXIM 5406	7
2.4 <i>Microphone</i>	8
2.5 Arduino Mega 2560	9
2.6 <i>Optocoupler Transistor</i>	10
2.7 <i>Power Supply</i>	10
2.8 Modul <i>Bluetooth</i>	11
2.9 <i>Arduino Software (IDE)</i>	12
2.10 <i>MIT App Inventor 2</i>	13
BAB III PERENCANAAN DAN PEMBUATAN ALAT	15
3.1 Blok Fungsional Sistem.....	15
3.2 Rancangan Mekanik	16
3.2.1 Perancangan Peletakan Rangkaian	17

3.2.2 Perancangan <i>Box</i> Penutup Rangkaian <i>Mixer</i>	19
3.3 Rancangan Elektrik	20
3.3.1 Rangkaian <i>Power Supply</i>	20
3.3.2 Rangkaian <i>Pre-Amplifier</i>	21
3.3.3 Rangkaian <i>Mixer</i>	23
3.3.4 Rangkaian MAXIM 5406 dengan <i>Push Button</i>	23
3.3.5 Shield untuk MAXIM 5406.....	24
3.3.6 Rangkaian MAXIM5406 dengan <i>Transistor Optocoupler</i> 4n25	25
3.3.7 Konfigurasi Arduino Mega 2560 dengan MAXIM5406....	26
3.3.8 Konfigurasi Arduino Mega 2560 dengan <i>Bluetooth</i>	27
3.4 Perancangan Perangkat Lunak	28
3.4.1 Perancangan Aplikasi <i>Android</i>	29
3.4.2 Perancangan <i>Software</i> Arduino.....	36
BAB IV PENGUJIAN DAN ANALISA DATA	39
4.1 Pengujian <i>Pre-Amp Channel 1</i>	39
4.2 Pengujian <i>Pre-Amp Channel 2</i>	41
4.3 Pengujian <i>Pre-Amp Channel 3</i>	44
4.4 Pengujian <i>Pre-Amp Channel 4</i>	47
4.5 Pengujian Rangkaian MAXIM5406 <i>Channel 1</i>	49
4.6 Pengujian Rangkaian MAXIM5406 <i>Channel 2</i>	51
4.7 Pengujian Rangkaian MAXIM5406 <i>Channel 3</i>	53
4.8 Pengujian Rangkaian MAXIM5406 <i>Channel 4</i>	55
4.9 Pengujian Rangkaian <i>Mixer</i>	58
4.10 Pengujian Status <i>Bluetooth</i> dan Pengiriman Data Terhadap Jarak Jangkauan <i>Bluetooth</i>	59
4.11 Pengujian Pengiriman Data <i>Android</i> ke Arduino Mega2560	62
BAB V PENUTUP	67
5.1 Kesimpulan	67
5.2 Saran.....	67
DAFTAR PUSTAKA	69
LAMPIRAN A	71
A.1. Tampilan Keseluruhan Rangkaian.....	71
A.2 Pengujian Pengiriman Data <i>Android</i> ke Arduino Mega 2560	76
LAMPIRAN B	105
B.1 <i>Listing</i> Program Arduino <i>Software</i>	105

B.2	<i>Listing Program MIT App Inventor 2</i>	111
LAMPIRAN C	125
C.1	<i>Data Sheet MAXIM 5406</i>	125
C.2	<i>Data Sheet Bluetooth HC-05</i>	137
C.3	<i>Data Sheet Arduino Mega 2560</i>	149
DAFTAR RIWAYAT HIDUP PENULIS 1	157
DAFTAR RIWAYAT HIDUP PENULIS 2	159

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 Konfigurasi Pin IC TL081CN	6
Gambar 2.2 MAXIM5406.....	8
Gambar 2.3 <i>Microphone</i>	9
Gambar 2.4 Arduino Mega 2560	9
Gambar 2.5 <i>Optocoupler Transistor</i>	10
Gambar 2.6 <i>Power Supply</i>	11
Gambar 2.7 Modul <i>Bluetooth</i>	12
Gambar 2.8 Tampilan <i>Arduino Software</i>	13
Gambar 2.9 Alur Pemindahan <i>Interface</i> dari <i>MIT App Inventor</i> ke <i>Android</i>	14
Gambar 3.1 Blok Fungsional Sistem.....	15
Gambar 3.2 Rancangan Lantai 1 Tampak Atas.....	18
Gambar 3.3 Rancangan Lantai 2 Tampak Atas.....	18
Gambar 3.4 Rancangan Lantai 1 dan Lantai 2 Tampak Atas.....	19
Gambar 3.5 Rancangan <i>Box</i> Penutup Tampak Samping.....	19
Gambar 3.6 Rancangan <i>Box</i> Penutup Tampak Depan.....	20
Gambar 3.7 Rangkaian <i>Power Supply</i>	21
Gambar 3.8 Rangkaian <i>Pre-Amplifier</i>	22
Gambar 3.9 Rangkaian <i>Mixer</i>	23
Gambar 3.10 Rangkaian MAXIM5406 dengan <i>Push Button</i>	24
Gambar 3.11 <i>Shield</i> untuk MAXIM5406.....	25
Gambar 3.12 Rangkaian MAXIM5406 dengan <i>Optocoupler 4n25</i>	26
Gambar 3.13 Konfigurasi Arduino Mega 2560 dengan <i>Bluetooth HC-</i> <i>05</i>	27
Gambar 3.14 Diagram Alur Proses Koneksi <i>Bluetooth</i>	28
Gambar 3.15 Tampilan <i>Design View</i>	29
Gambar 3.16 Tampilan Menu Utama Aplikasi	30
Gambar 3.17 Tampilan Menu pada <i>Channel</i>	31
Gambar 3.18 <i>Block</i> Program Konektivitas <i>Bluetooth</i>	33
Gambar 3.19 <i>Block</i> Program Status Koneksi <i>Bluetooth</i>	33
Gambar 3.20 <i>Block</i> Program Tombol Tu Ketika di <i>TouchDown</i>	34
Gambar 3.21 <i>Block</i> Program Ketika Tombol Tu di <i>Touch Up</i>	35
Gambar 3.22 <i>Block</i> Program Inisialisasi <i>LayOut</i>	36
Gambar 3.23 Deklarasi Nilai <i>Char</i>	37
Gambar 3.24 Fungsi <i>Setup</i>	37

Gambar 3.25 Kecepatan Transmisi Data	37
Gambar 3.26 Eksekusi Program	37
Gambar 3.27 Proses Data	38
Gambar 4.1 Pengujian <i>Pre-Amp Channel 1</i>	39
Gambar 4.2 Pengujian <i>Input dan Output Pre-Amp Channel 1</i>	40
Gambar 4.3 Gelombang <i>Input dan Output Pre-Amp channel 1</i>	41
Gambar 4.4 Pengujian <i>Pre-Amp Channel 2</i>	42
Gambar 4.5 Pengujian <i>Input dan Output Pre-Amp Channel 2</i>	43
Gambar 4.6 Gelombang <i>Input dan Output Pre-Amp Channel 2</i>	43
Gambar 4.7 Pengujian <i>Pre-Amp Channel 3</i>	44
Gambar 4.8 Pengujian <i>Input dan Output Pre-Amp Channel 3</i>	45
Gambar 4.9 Gelombang <i>Input dan Output Pre-Amp Channel 3</i>	46
Gambar 4.10 Pengujian <i>Pre-Amp Channel 4</i>	47
Gambar 4.11 Pengujian <i>Input dan Output Pre-Amp Channel 4</i>	48
Gambar 4.12 Gelombang <i>Input dan Output Pre-Amp Channel 4</i>	48
Gambar 4.13 <i>Tone Control Digital MAXIM5406 Channel 1</i>	49
Gambar 4.14 <i>Input Output MAXIM5406 Channel 1</i>	50
Gambar 4.15 <i>Tone Control Digital MAXIM5406 Channel 2</i>	51
Gambar 4.16 <i>Input Output MAXIM5406 Channel 2</i>	52
Gambar 4.17 <i>Tone Control Digital MAXIM5406 Channel 3</i>	53
Gambar 4.18 <i>Input Output MAXIM5406 Channel 3</i>	54
Gambar 4.19 <i>Tone Control Digital MAXIM5406 Channel 4</i>	56
Gambar 4.20 <i>Input Output MAXIM 5406 Channel 4</i>	57
Gambar 4.21 Rangkaian <i>Mixer</i>	58
Gambar 4.22 Pengujian Jarak Jangkauan <i>Bluetooth</i> dengan Halangan	60
Gambar 4.23 Pengujian Jarak Jangkauan <i>Bluetooth</i> Tanpa Halangan	61
Gambar 4.24 Tampilan Aplikasi Ketika Terhubung	62
Gambar 4.25 Data yang Akan Dikirim oleh Aplikasi <i>Android</i>	63
Gambar 4.26 Program Untuk Mengirim Data “A”	63
Gambar 4.27 Data yang Dibaca oleh Arduino Mega 2560	64

DAFTAR TABEL

HALAMAN

Tabel 3.1	Daftar <i>Wiring</i> Arduino Mega 2560 dengan MAXIM5406...	27
Tabel 3.2	Penjelasan <i>Icon</i> pada Aplikasi <i>Android</i>	31
Tabel 3.3	Data yang Dikirim ke Arduino Mega 2560	34
Tabel 4.1	Spesifikasi Rangkaian <i>Pre-Amp Channel 1</i>	40
Tabel 4.2	Spesifikasi Rangkaian <i>Pre-Amp Channel 2</i>	42
Tabel 4.3	Spesifikasi Rangkaian <i>Pre-Amp Channel 3</i>	45
Tabel 4.4	Spesifikasi Rangkaian <i>Pre-Amp Channel 4</i>	47
Tabel 4.5	Spesifikasi Rangkaian MAXIM5406 <i>Channel 1</i>	50
Tabel 4.6	Spesifikasi Rangkaian MAXIM5406 <i>Channel 2</i>	52
Tabel 4.7	Spesifikasi Rangkaian MAXIM5406 <i>Channel 3</i>	54
Tabel 4.8	Spesifikasi Rangkaian MAXIM5406 <i>Channel 4</i>	56
Tabel 4.9	Spesifikasi Rangkaian <i>Mixer</i>	59
Tabel 4.10	Pengujian dengan Halangan	60
Tabel 4.11	Pengujian Tanpa Halangan	61
Tabel 4.12	Data Teks yang Dibaca oleh Arduino	64

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Akhir ini perkembangan teknologi berkembang sangat inovatif terutama dalam hal *sound engineer*. *Mixer* adalah salah satu perangkat paling populer setelah *microphone*. Di kehidupan sehari-hari lebih dikenal dengan sebutan *mixer*, mungkin kebanyakan orang menyebutnya demikian karena fungsinya yang memang mencampur segala suara yang masuk, kemudian menyeimbangkannya, menjadikannya salura dua kanal (L-R kalau stereo, dan satu kalau mono) atau mengatur suara dari suatu *microphone channel* atau yang disebut *tone control* digital. Pada umumnya masih sering ditemui seorang *sound engineer* secara manual mengatur potensio geser pada *mixer* audio namun dalam monitoring keluaran yang dihasilkan *mixer* telah banyak menggunakan monitor digital.

Seorang operator masih melakukan secara manual dalam men-*setting mixer* audio. Pengaturan *mixer* audio tersebut secara manual yaitu mengatur *mixer* audio dimana *mixer* tersebut ditata dan harus memutar tombol / potensio tanpa dapat berpindah-pindah tempat untuk mendengarkan dan menyesuaikan secara langsung hasil pengaturan *mixer* yang diatur ditempat yang diinginkan. Hal tersebut dirasa kurang praktis. Melihat kasus ini dibuatlah inovasi baru dengan membuat suatu aplikasi berbasis *android* yang kita hubungkan pada rangkaian *tone control* menggunakan fasilitas *Bluetooth*, dimana didalam aplikasi tersebut terdapat opsi menu *Volume*, *Bass*, *Treble*, dan *Mute* seperti tampilan pada umumnya suatu audio *mixer*. Dengan hanya menggunakan perangkat *android* diharapkan para operator/*sound engineer* dapat mengatur *mixer* audio secara praktis yang tidak hanya terbatas oleh suatu jarak.

1.2 Permasalahan

Pada umumnya pengaturan *mixer* audio yang dilakukan oleh seorang *sound engineer* masih manual, artinya *sound engineer* harus melakukan pengaturan *mixer* audio tersebut yaitu mengatur *mixer* audio dimana *mixer* tersebut ditata dan harus memutar tombol secara menetap tanpa dapat berpindah-pindah tempat untuk mendengarkan dan menyesuaikan secara nyata hasil pengaturan *mixer* yang diatur di tempat

yang disesuaikan dan diperlukan. Hal tersebut dirasa kurang praktis dan kurang efisien. Oleh sebab itu, diciptakan suatu inovasi baru dalam pengaturan *mixer* audio secara praktis dengan menggunakan *android* dengan menggunakan komunikasi *via Bluetooth* sehingga *tone control* dapat dilakukan dari jarak yang lebih jauh.

1.3 Batasan Masalah

Batasan masalah dalam Tugas Akhir ini diantaranya adalah:

1. *Mixer* Audio ini menyediakan 4 *Channel Input*.
2. Opsi kendali / *setting* audio yang tersedia *Volume, Bass, Middle, Trebble*, dan *Mute*
3. Jarak komunikasi kurang lebih 15 meter karena menggunakan *Bluetooth*

1.4 Tujuan

Topik Tugas Akhir ini bertujuan merancang/membuat suatu aplikasi kontrol *mixer* audio berbasis *Android* dengan *via Bluetooth* guna mengatur *Volume, Treble, Bass*, dan *Mute*. Dari uraian tersebut, maka dapat dibagi menjadi tiga tujuan dalam proyek akhir ini, yaitu:

1. Merancang/membuat suatu perangkat keras *mixer* audio berbasis *android via Bluetooth* guna mengatur *Volume, Trebble, Bass* dan *Mute* yang dapat diatur tanpa harus ditempatkan dimana *mixer* audio terpasang (Penanggung Jawab Nafiqul Ihsan)
2. Membuat perangkat lunak *tone control* digital yang dilakukan melalui *Bluetooth* (Penanggung jawab Dyah Ayuningtyas)

1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, tahap persiapan yang dilakukan dengan cara pencarian data, bahan, dan literatur. Yang kedua yaitu perencanaan dan pembuatan alat yang terdiri dari dua yaitu *hardware* dan *software*. Yang ketiga yaitu tahap uji dan analisa dan tahap yang terakhir adalah tahap penyusunan laporan.

Pada tahap persiapan yaitu studi literatur akan dipelajari mengenai konsep dari *Arduino Mega 2560*, modul *Bluetooth*, *MAXIM5406*, *optocoupler transistor*, dan pemrograman yang digunakan yaitu *software Arduino* dan *MIT App Inventor 2*. Pada tahap perencanaan dan pembuatan alat terdiri dari dua yaitu, perancangan dan pembuatan perangkat keras dan perencanaan dan pembuatan perangkat lunak. Perancangan dan pembuatan perangkat keras terdiri dari pembuatan modul *MAXIM5406* yang digabungkan dengan *optocoupler transistor*

pada setiap bagian pin *Volume*, *Bass*, *Treble*, dan *Mute*. Kemudian perancangan dan pembuatan perangkat lunak terdiri dari pembuatan *interface* di *android* dengan menggunakan *MIT App Inventor 2* dan pemrograman dengan *software* *Arduino* untuk komunikasi yang dilakukan antara modul *Bluetooth* dengan MAXIM5406. Tahap selanjutnya adalah tahap uji dan analisa yaitu dengan hasil akhir mendapatkan ketepatan system transmisi data menggunakan komunikasi modul *Bluetooth*. Data percobaan yang telah diperoleh selanjutnya akan dianalisis. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari Rangkaian *Pre-Amp*, TL081CN, MAXIM5406, *Microphone*, *Arduino Mega 2560*, *optocoupler 4n25*, *power supply* yang merupakan tanggungjawab dari Nafiqul Ihsan dan modul *Bluetooth HC-05*, *Arduino Software (IDE)*, dan *MIT App Inventor 2* yang merupakan tanggungjawab dari Dyah Ayuningtyas.

Bab III Perencanaan dan Pembuatan Alat

Bab ini membahas perancangan perangkat keras dan perangkat lunak *tone control mixer* audio berdasarkan teori dasar pada Bab II

Bab IV Pengujian dan Analisa Data

Bab ini memuat hasil simulasi dan implementasi serta analisis dari hasil pengujian perangkat keras dan perangkat lunak.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Diharapkan dengan Tugas Akhir ini dapat memberikan sumbangsih pemikiran, mempermudah operator atau *sound engineer* dalam pengaturan *mixer* audio dengan jarak yang lebih jauh sehingga mendapatkan audio dengan jernih dan sesuai pada porsinya dan sesuai yang diinginkan.

BAB II

TEORI DASAR

Pada bab ini akan dibahas mengenai tinjauan pustaka dimana ada 10 poin. Tinjauan pustaka ini dibagi menjadi 2 bagian tiap masing masing mahasiswa, Nafiqul Ihsan mencari bahan dari materi pada poin 2.2 sampai dengan poin 2.7. dan Dyah Ayunintyas mencari bahan dari materi pada poin 2.8 sampai dengan poin 2.10. dimana tiap materi yang dicari sebagai dasar materi untuk pengerjaan alat yang dibuat masing-masing mahasiswa, dan juga sebagai dasar materi untuk pembuatan keseluruhan alat ini.

2.1 Tinjauan Pustaka [1]

Ada dua sistim *mixer* yang sering dijumpai, yaitu yang pertama sistem *mixer* yang dikontrol secara manual dengan menggunakan tangan dimana sensor yang digunakan adalah telinga, kemudian yang kedua adalah system *mixer* yang dikontrol secara digital yang biasanya dipadukan dengan smartphone. Sistem digital ini dirancang agar dapat mengontrol suatu *mixer* dengan jarak tertentu serta tidak harus di tempat dimana *mixer* itu dipasang. Sebelum masuk ke rangkian *mixer* maka ada beberapa hal yang harus dipenuhi yaitu rangkain *Pre-Amp*, *tone control*, serta catu daya untuk masing masing rangkaian. Masukan dari rangkaian *Pre-Amp* adalah suatu sinyal audio yang dapat diciptakan oleh perangkat *microphone* , gitar, *orgen* , dan alat-alat instrument lainnya.

Diantara blok rangkaian penguat depan dengan penguat akhir terdapat blok penguat pengatur. Dalam penguat pengatur ini terdapat pengaturan kuat suara, pengaturan nada dan pengatur kesetimbangan kanal untuk sistem stereo. Pengatur kuat suara berfungsi menyesuaikan kuat suara sekeliling dengan kebiasaan mendengar. Sedang pengatur nada untuk menyesuaikan dengan akustik ruangan.[1]

Telah dibuat rangkaian *Pre-Amp* yang menggunakan IC TL081CN yang merupakan IC yang berguna sebagai *Oph-Amp*, dimana tugas dari IC ini adalah membesarkan nilai tegangan dari sinyal masukan (*Audio Signal*) guna dapat diterima oleh rangkaian *tone control*. Rangkaian *tone control* ini menggunakan MAXIM5406 dimana pada IC tersebut telah terdapat rangkaian *Tone control* yang meliputi *Bass up*, *Bass down*, *Treble up*, *Treble down*, *Volume up*, serta *Volume down*. Dengan adanya konfigurasi *tone control* tersebut kemudian dikontrol secara digital menggunakan Arduino Mega 2560 yang disambungkan dengan perangkat *android* yang berfungsi sebagai *controller*. Sinyal keluaran

dari rangkaian *tone control*, masuk ke rangkaian *mixer* guna menyatukan beberapa *channel audio*, yang bertanggung jawab adalah Nafiqul Ihsan.

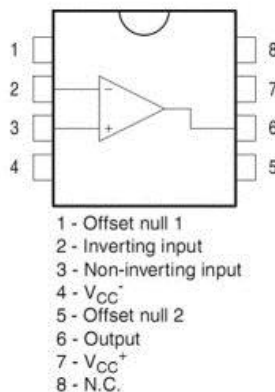
Pada Tugas Akhir ini akan dilakukan perancangan kontrol *mixer audio* secara digital yang di padukan dengan perangkat *android* sebagai *tools* untuk melakukan kontrol. Dimana dalam menghubungkan antara Arduino Mega 2560 dengan *smartphone android* menggunakan modul *Bluetooth HC-05* agar dapat kontrol secara jarak jauh, serta dalam proses pembuatan *software android* , menggunakan suatu perangkat lunak yaitu *MIT App Inventor 2*.

2.2 IC TL081CN [2]

Setiap sinyal keluaran dari audio memiliki *output* tegangan yang relatif kecil, maka dari itu kita membutuhkan suatu rangkaian *Pre-Amp* yang menggunakan suatu IC *operational amplifier* guna dapat memberikan penguatan beberapa kali dari tegangan masukannya.

TL081CN adalah penguat operasi J-FET berkecepatan tinggi yang memiliki satu catu tegangan 12 V dc. J-FET dan *transistor bipolar* dalam rangkaian terpadu *monolitik*. Perangkat ini memiliki bias masukan rendah dan arus *offset* rendah.[2]

IC TL081CN memiliki 8 pin dimana setiap konfigurasi pin memiliki fungsinya masing-masing. Jenis IC ini hanya memiliki 1 *Oph-Amp* didalam ICnya. Ada pula IC jenis ini yang memiliki banyak *Oph-Amp* didalam satu IC, untuk lebih jelasnya mengenai konfigurasi setiap pin dari IC TL081CN dapat dilihat pada Gambar 2.1. Konfigurasi Pin IC TL081CN. Yang bertanggung jawab adalah Nafiqul Ihsan.



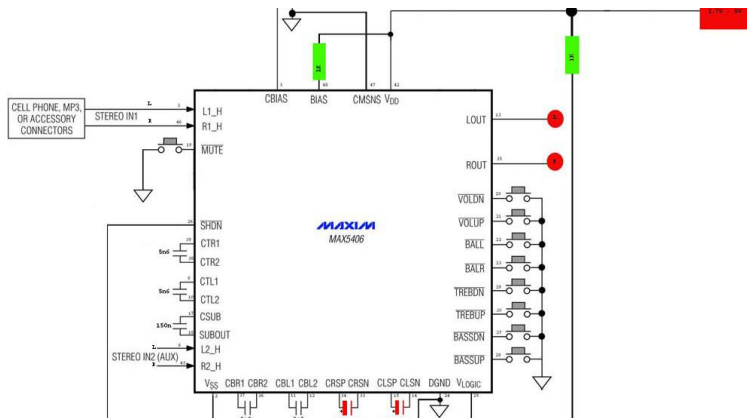
Gambar 2.1 Konfigurasi Pin IC TL081CN

Berdasarkan gambar diatas, dapat diketahui fungsi dari masing-masing kaki pin konfigurasi dimana dalam penerapannya terdapat rangkain *Pre-Amp inverting* maupun *non-inverting* dimana letak perbedaanya terdapat pada *feedback* ke *output*. Jika *feedback* berasal dari pin 2 maka disebut *inverting Pre-Amp*, kemudian jika *feedback* dari pin 3 maka disebut *Non-Inverting Pre-Amp*. Selain itu untuk pin 4 merupakan catu daya yang bernilai negatif, sedangkan untuk catu daya positif terdapat pada pin 7. *Input* tegangan untuk IC ini sebesar 12 Vdc.

2.3 MAXIM 5406 [3]

MAXIM5406 prosesor audio stereo menyediakan solusi audio yang lengkap dengan *Volume*, *Bass*, *Treble*, dan *Mute*. Ini fitur *dual 32-tap* potensiometer logaritmik untuk kontrol *Volume*, potensiometer ganda untuk kontrol keseimbangan, dan potensiometer digital linear untuk kontrol nada. Sebuah sederhana antarmuka tombol tekan *debounced* mengontrol semua fungsi. MAXIM5406 kemajuan *wiper* pengaturan sekali per *push button*. MeMegang kendali *input* rendah selama lebih dari 1s kemajuan *wiper* pada tingkat 4 Hz untuk 4s dan 16 Hz setelahnya. Sebuah fitur penekanan klik/pop terintegrasi menghilangkan kebisingan suara yang dihasilkan oleh gerakan *wiper* ini. Yang bertanggung jawab adalah Nafiqul Ihsan.

MAXIM5406 menyediakan *output subwoofer* yang *internal* menggabungkan saluran kiri dan kanan. *Filter* kapasitor *eksternal* memungkinkan untuk frekuensi *cut-off* disesuaikan untuk *output subwoofer*. Sebuah modus *Bass-boost* meningkatkan respon frekuensi rendah dari saluran kiri dan kanan. Bias *amplifier* terintegrasi menghasilkan $(VDD + VSS) / 2$ tegangan bias yang diperlukan, menghilangkan kebutuhan untuk *Op-Amp* eksternal untuk operasi *unipolar*. Seperti pada Gambar 2.2 MAXIM5406 juga dilengkapi kontrol suasana untuk meningkatkan pemisahan kiri dan kanan saluran *output* untuk *headphone* dan sistem *speaker desktop*, dan fitur *pseudostereo* yang mendekati suara stereo dari sinyal *monophonic*. [3]



Gambar 2.2 MAXIM5406

2.4 Microphone [4]

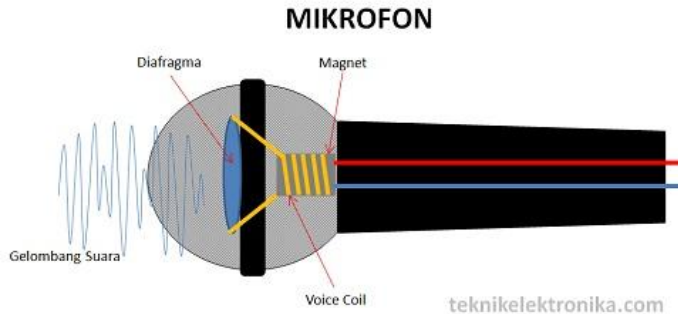
Microphone atau Mikrofon merupakan komponen penting dalam perangkat Elektronik seperti alat bantu pendengaran, perekam suara, penyiaran Radio maupun alat komunikasi lainnya seperti *Handphone*, Telepon, *Interkom*, *Walkie Talkie* serta *Home Entertainment* seperti Karaoke. Pada dasarnya sinyal listrik yang dihasilkan *Microphone* sangatlah rendah, oleh karena itu diperlukan penguat sinyal yang biasanya disebut dengan *Amplifier*.

Pada mikrofon dinamik tegangan bolak-balik dibangkitkan melalui induksi yang sebanding dengan kecepatan membran. Gelombang bunyi menggetarkan membran maka kumparan akan bergerak tegak lurus terhadap arah medan.[4]

Mikrofon merupakan salah satu *transduser* (perangkat yang mengubah energi dari satu bentuk ke bentuk lainnya) dimana mikrofon mengubah energi akustik (gelombang suara) menjadi energi listrik (sinyal audio). Ada berbagai tipe mikrofon dimana masing-masing tipe menggunakan metode yang berbeda dalam mengkonversi energi, namun semua tipe mikrofon tersebut memiliki satu kesamaan yaitu diafragma. Diafragma merupakan sebuah material tipis (berupa kertas, plastik atau alumunium) yang bergetar ketika terkena gelombang suara.

Pada Tugas Akhir ini *microphone* digunakan sebagai *input audio* yang masuk ke rangkaian *Pre-Amp* sebelum menuju ke rangkaian *tone control*. Untuk lebih jelasnya, Gambar 2.3 *microphone* merupakan

bagian dalam dari *microphone*. Yang bertanggung jawab mencari bahan adalah Nafiqul Ihsan.



Gambar 2.3 *Microphone*

2.5 Arduino Mega 2560 [5]

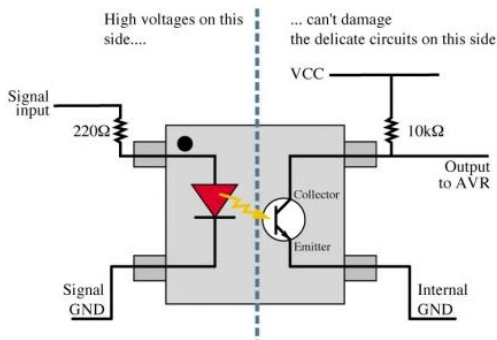
Arduino Mega 2560 adalah papan mikrokontroler yang berbasis ATMega2560 . Ini memiliki 54 pin *input / output* digital (15 dapat digunakan sebagai *output* PWM), 16 *input analog*, 4 UART (*port serial* perangkat keras), *osilator kristal* 16 MHz, koneksi USB, colokan listrik, *header ICSP*, dan tombol *reset*. Arduino Mega 2560 *ditunjukkan* seperti pada Gambar 2.4. Ini berisi semua yang dibutuhkan untuk mendukung mikrokontroler. Cukup hubungkan ke komputer dengan kabel USB atau nyalakan dengan adaptor AC ke DC catu baterai untuk memulai.[5]. Yang bertanggung jawab adalah Nafiqul Ihsan.



Gambar 2.4 Arduino Mega 2560

2.6 Optocoupler Transistor [6]

Optocoupler adalah suatu piranti yang terdiri dari 2 bagian yaitu *transmitter* dan *receiver*, yaitu antara bagian cahaya dengan bagian deteksi sumber cahaya terpisah. Biasanya *optocoupler* digunakan sebagai saklar elektrik, yang bekerja secara otomatis. *Optocoupler* atau *optoisolator* merupakan komponen penggandeng (*coupling*) antara rangkaian *input* dengan rangkaian *output* yang menggunakan media cahaya (*opto*) sebagai penghubung seperti yang ditampilkan pada Gambar 2.5.[6]. Yang bertanggung jawab adalah Nafiqul Ihsan.



Gambar 2.5 *Optocoupler Transistor*

2.7 Power Supply [7]

Catu daya sebagai sumber tenaga dapat berasal dari baterai, *accu*, *solar cell*, dan adaptor. Komponen ini akan mencatu tegangan sesuai dengan tegangan yang diperlukan oleh rangkaian elektronika.[4]

Bagian catu daya yang dibuat terdiri dari komponen baterai dan komponen regulator tegangan seri LM78XX. Seri regulator 78XX (7805, 7808, atau 7012) adalah regulator tegangan tiga terminal tipikal. Regulator yang digunakan adalah LM7805 yang menghasilkan tegangan 5 V.[7]

Power supply merupakan perangkat keras yang mampu menyuplai tegangan listrik secara langsung dari sumber tegangan listrik ke perangkat yang membutuhkan tegangan listrik. *Power supply* memiliki *input* dari tegangan yang berarus AC dan mengubahnya menjadi arus DC lalu menyalurkannya ke berbagai perangkat keras yang membutuhkannya. Karena arus DC yang dibutuhkan untuk perangkat keras agar dapat beroperasi, arus DC bisa disebut juga sebagai arus yang searah, sedangkan arus AC merupakan arus yang berlawanan. *Power*

Supply merupakan komponen yang sangat penting agar perangkat keras yang digunakan bisa berjalan dengan baik dan optimal. Tegangan keluaran *power supply* yang dibutuhkan dan digunakan pada perangkat keras biasanya 24 V, 12 V, 9 V, dan 5 V. *Power Supply* yang digunakan dapat dilihat pada Gambar 2.6 Yang bertanggung jawab membuat rangkaian ini adalah Nafiqul Ihsan.



Gambar 2.6 *Power Supply*

2.8 Modul Bluetooth [8]

Modul *Bluetooth* HC-05 merupakan modul komunikasi nirkabel pada frekuensi 2,4 GHz dengan pilihan koneksi bisa sebagai *slave*, ataupun sebagai *master*. Sangat mudah digunakan dengan mikrokontroler untuk membuat aplikasi *wireless*. *Interface* yang digunakan adalah serial RXD, TXD, VCC dan GND. *Built in LED* sebagai indikator koneksi *Bluetooth*. Tegangan *input* antara 3,6-6 V, 4 *pin interface* 3,3 V dapat langsung dihubungkan ke berbagai macam mikrokontroler (khusus Arduino, 8051, 8535, AVR, PIC, ARM, MSP430.). Modul *Bluetooth* yang digunakan seperti pada Gambar 2.7. Jarak efektif jangkauan sebesar 10 meter, meskipun dapat mencapai lebih dari 10 meter, namun kualitas koneksi makin berkurang.[8]. Yang bertanggung jawab adalah Dyah Ayuningtyas.

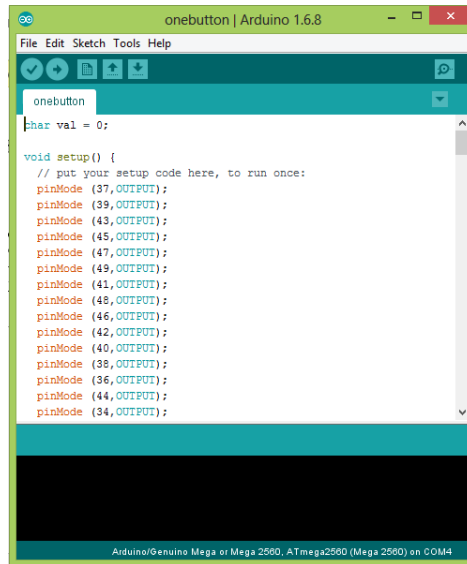


Gambar 2.7 Modul *Bluetooth*

2.9 Arduino Software (IDE) [9]

Arduino software bersifat *IDE* yang merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler. Yang bertanggung jawab membuat program ini adalah Dyah Ayuningtyas.

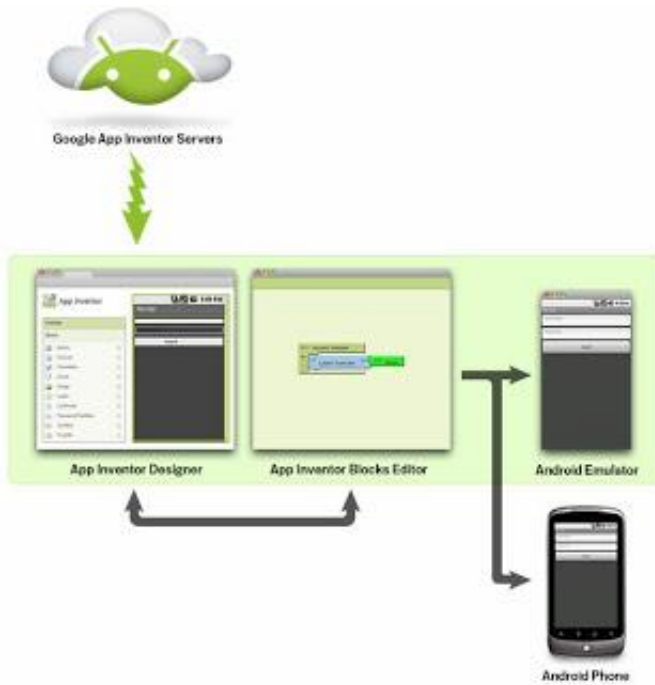
Arduino *IDE* dibuat dari bahasa pemrograman JAVA. Arduino *IDE* juga dilengkapi dengan *library* C/C++ yang biasa disebut *wiring* yang membuat operasi *input* dan output menjadi lebih mudah. Arduino *IDE* ini dikembangkan dari *software Processing* yang dirombak menjadi Arduino *IDE* khusus untuk pemrograman dengan Arduino pada Gambar 2.8 ditunjukkan tampilan Arduino *software*. Yang bertanggungjawab adalah Dyah Ayuningtyas.



Gambar 2.8 Tampilan *Arduino Software*

2.10 MIT App Inventor 2 [10]

Saat ini untuk membuat sebuah aplikasi berbasis *android* sangatlah mudah, salah satunya dengan *MIT App Inventor 2* merupakan sebuah *tool* untuk membuat aplikasi *android*, yang menyenangkan dari *tool* ini adalah karena berbasis visual *block programming* atau antarmuka grafis, sehingga bisa membuat aplikasi tanpa kode satupun. Disebut visual *block programming* karena manusia akan melihat, menggunakan, menyusun dan *drag-drops* “blok” yang merupakan simbol-simbol perintah dan fungsi tertentu dalam membuat aplikasi, dan secara sederhana dengan bisa menyebutnya tanpa menuliskan kode program. Pada Gambar 2.9 ditunjukkan alur pemindahan *interface* yang dibuat pada *MIT App Inventor 2* ke *Android*. [10]. Yang bertanggung jawab adalah Dyah Ayuningtyas.



Gambar 2.9 Alur Pemindahan *Interface* dari *MIT App Inventor* ke *Android*

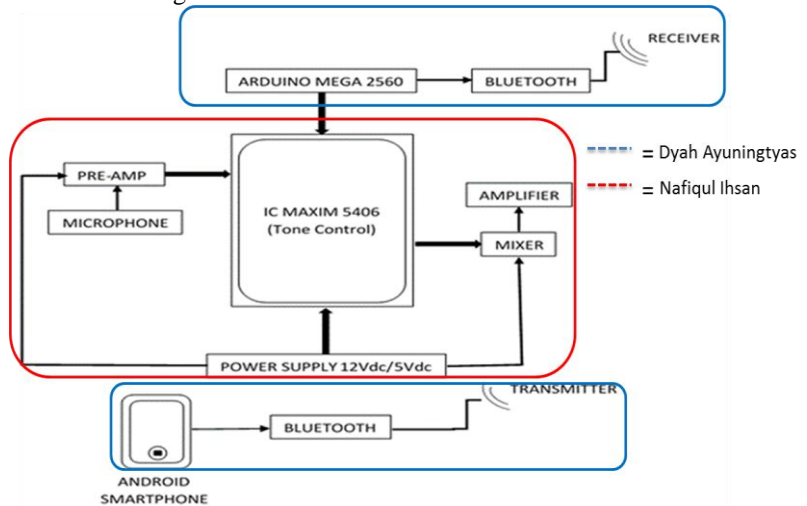
BAB III

PERENCANAAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas mengenai perancangan sistem dari alat yang akan dibuat. Dimana akan dibagi menjadi dua bagian, untuk Nafiqul Ihsan merancang serta membuat rangkaian *pre-amp*, *tone control* digital, dan rangkaian *mixer*. Untuk Dyah Ayuningtyas bertugas untuk membuat *software* android yang menggunakan aplikasi *MIT App Inventor 2* dan melakukan *wiring* terhadap Arduino Mega 2560 ke rangkaian MAXIM5406.

3.1 Blok Fungsional Sistem

Sebelum melakukan perancangan perangkat keras dan perangkat lunak, diperlukan sebuah perancangan blok fungsional sistem berupa blok diagram yang menjelaskan sistem kerja secara keseluruhan. Tugas Akhir ini. Secara keseluruhan blok fungsional sistem dapat dilihat pada Gambar 3.1. Blok Fungsional Sistem.



Gambar 3.1 Blok Fungsional Sistem

Sesuai dengan Gambar 3.1, dijelaskan tentang tindak kontrol terhadap MAXIM5406 yang berperan sebagai *tone control*. Berawal dari rangkaian *Pre-Amp* yang terdiri dari 4 *chanel* yang diberi sinyal audio

berupa *microphone*, gitar dan lain sebagainya. Fungsi dari rangkaian *Pre-Amp* ini berguna untuk membesarkan beberapa kali teangan *input* sinyal audio supaya dapat di terima oleh rangkaian MAXIM5406. Hal ini dilakukan karena tegangan *output* yang dikeluarkan oleh *microphone* masih terlalu kecil, sehingga perlu dilakukan pembesaran hingga beberapa kali. Untuk setiap rangkaian *Pre-Amp*, MAXIM5406, dan *mixer* membutuhkan catu daya dari *power supply* sebesar 12 V,5 V, dan 12 V.

Selain itu terdapat juga rangkaian *mixer* yang terhubung ke *amplifier*. Rangkaian *mixer* berfungsi untuk menggabungkan beberapa output audio dari MAXIM5406 untuk di jadikan satu kemudian dibesarkan nilainya tegangannya menggunakan *amplifier*. Kemudian MAXIM5406 akan dikontrol secara digital dengan perantara *transistor optocoupler* menggunakan *smartphone android*. Dalam proses pembuatan *software android*, menggunakan aplikasi yang bernama *MIT App Inventor 2* yang dapat di kirim langsung ke *smarthphone* menggunakan akun.

Kemudian terdapat juga Arduino Mega 2560 guna mengontrol perubahan settingan tone yang dikirim oleh *smartphone android*. Arduino Mega 2560 dihubungkan langsung dengan modul *Bluetooth HC-05* yang berperan sebagai *receiver* dari sistem, dan yang berperan sebagai *transmitter* adalah *smartphone android* yang terhubung dengan *Bluetooth* guna mengirim data kepada *receiver*.

Bentuk tampilan dari *software* pada *smartphone android* cukup sederhana, yaitu terdapat menu *connect to Bluetooth* dan pilihan untuk setiap *channel*nya seperti *Volume up/down*, *Bass up/down*, *Treble up/down*, serta *Mute*. Dalam proses komunikasi data untuk *receiver* menggunakan modul *Bluetooth* pabrikan yaitu dengan tipe HC-05 serta untuk *transmitter* yang dikirim oleh *smartphone android* ,menggunakan fasilitas *Bluetooth* yang ada pada *smartphone android*.

Untuk sistem diatas merupakan serangkaian *loop* tertutup dimana untuk sensor *feedback*nya menggunakan telinga dari pendengar agar dapat di sesuaikan dengan selera suara yang diinginkan oleh seorang *sound engineer*.

3.2 Rancangan Mekanik

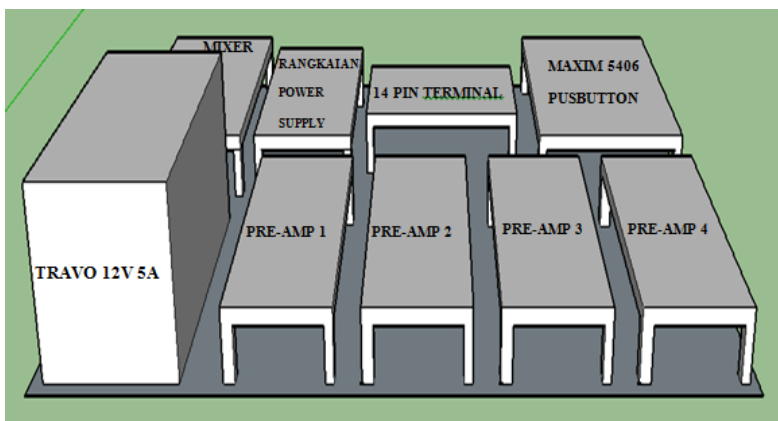
Pada Sub bab ini akan dibahas mengenai perancangan mekanik untuk tugas ini. Yang bertanggung jawab untuk rancangan mekanik adalah Nafiqul Ihsan. Perancangan mekanik berupa perancangan

perangkat keras yang mendukung seluruh perancangan dan pembuatan alat. Perancangan mekanik yang akan dibahas meliputi perancangan peletakan rangkaian dari *Pre-Amp*, *mixer*, *tone control*, *powersupply* dan Arduino serta desain *box* luar menutup keseluruhan rangkain *mixer* yang digunakan untuk Tugas Akhir ini. Untuk peletakan perancangan, dibagi menjadi 2 lantai dimana terdapat lantai 1 serta lantai 2 yang dihubungkan dengan *double speacer*. Untuk komponen yang terpasang pada lantai satu meliputi rangkaian *Power Supply*, rangkaian *Pre-Amp*, rangkaian terminal dengan total terdapat 14 pin terminal *block*, rangkaian *mixer* serta yang terakhir adalah rangkaian MAXIM5406 dengan *push button interface*. Sedangkan untuk konfigurasi lantai 2 pada desain mekanik kami yaitu meliputi, rangkaian *tone control* MAXIM 5406 dengan *transistor optocoupler 4n25* sebanyak 4 *channel*, serta Arduino yang telah dilengkapi *shield* khusus untuk MAXIM5406 dan Modul *Bluetooth* yang tertancap pada Arduino Mega 2560.

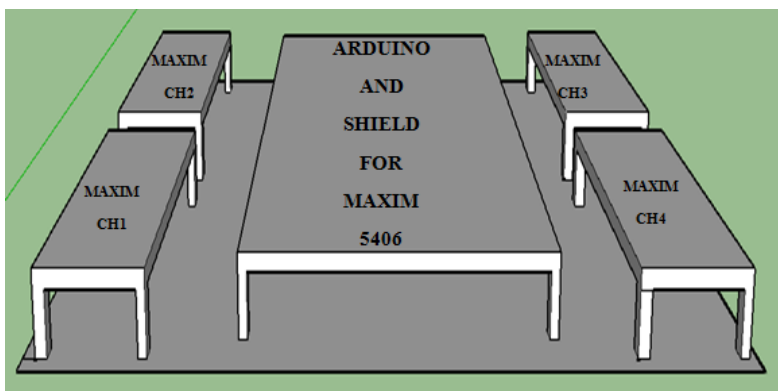
3.2.1 Perancangan Peletakan Rangkaian

Untuk peletakan setiap rangkaian, dibagi menjadi 2 tingkat yaitu bagian atas yang ditunjukkan pada Gambar 3.2 dan bagian bawah yang ditunjukkan pada Gambar 3.3 dihubungkan dengan *specier* besi ukuran sedang. Untuk yang bagian bawah yang berukuran sekitar 43 cm x 20 cm terdapat rangkaian *power supply* yang terdapat pada sisi ujung kemudian disebelahnya diikuti oleh 4 rangkaian *Pre-Amp*, 1 rangkaian *mixer*, dan 1 rangkaian MAXIM5406 dengan *Push button*. Serta terdapat beberapa terminal blok guna memperbanyak ruang untuk *power supply*. Yang bertanggung jawab adalah Nafiqul Ihsan.

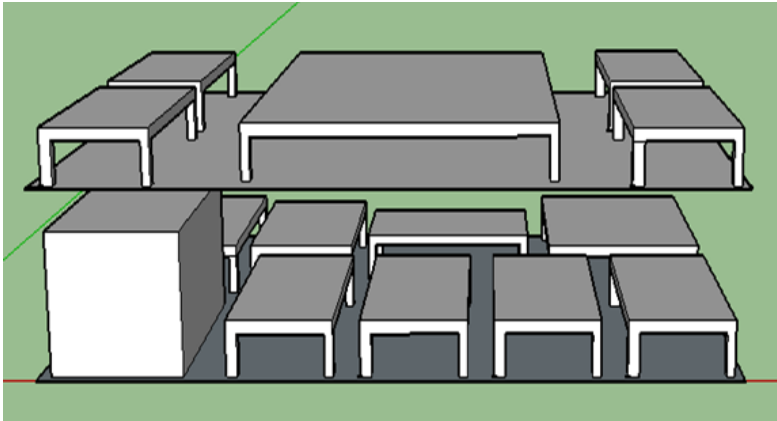
Pada bagian atas terdapat rangkaian MAXIM5406 dengan *transistor optocoupler 4n25* sebanyak 4 dan Arduino Mega2560 yang telah dilengkapi *shield* serta *Bluetooth*, dan untuk *wiring* digunakan kabel serabut biasa untuk menghubungkan antara bagian atas dengan bagian bawah yang ditunjukkan pada Gambar 3.4.



Gambar 3.2 Rancangan Lantai 1 Tampak Atas



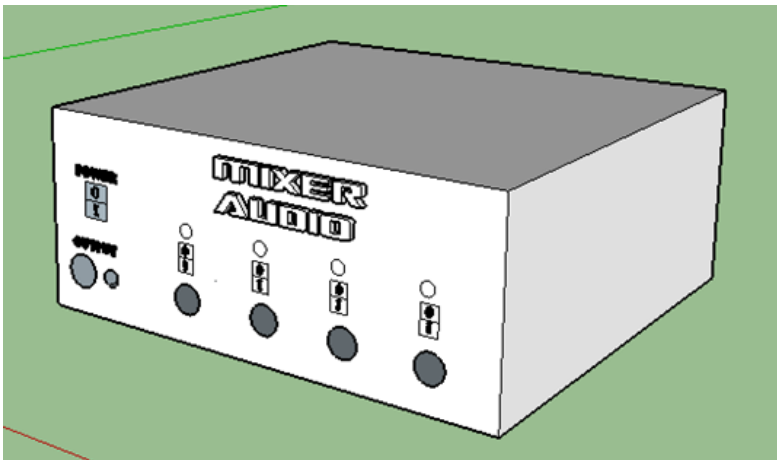
Gambar 3.3 Rancangan Lantai 2 Tampak Atas



Gambar 3.4 Rancangan Lantai 1 dan Lantai 2 Tampak Atas

3.2.2 Perancangan *Box* Penutup Rangkaian *Mixer*

Untuk perancangan *box* luar digunakan akrilik yang berbentuk seperti balok dengan dimensi tertentu dan dilengkapi lubang guna memasukan semua kabel *jack input* untuk *mic1*, *mic2*, gitar dan *orgen*. Serta beberapa *switch on/off* sebagai indicator catu daya seperti yang ditunjukkan pada Gambar 3.5. Yang bertanggung jawab dalam perancangan *box* penutup rangkaian *mixer* adalah Nafiqu Ihsan.



Gambar 3.5 Rancangan *Box* Penutup Tampak Samping



Gambar 3.6 Rancangan *Box* Penutup Tampak Depan

Untuk tutup bagian atas digunakan akrilik biasa yang menggunakan engsel yang terbuat dari akrilik. Didesain sehingga rangkaian dalam akan mudah untuk dikeluarkan. Untuk setiap *channel* diberi *switch* guna untuk memutus maupun menyambung *channel* yang ingin digunakan serta memberi indikator led 5 mm berwarna merah sebagai tanda aktif atau tidaknya *channel* tersebut seperti yang telah ditunjukkan pada Gambar 3.6.

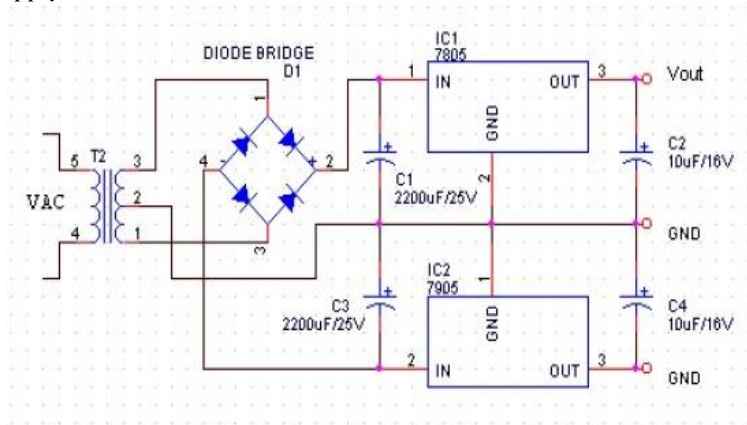
3.3 Rancangan Elektrik

Pada Sub bab perancangan elektrik dibahas tentang rangkaian elektrik beserta komponen – komponen yang digunakan dalam Tugas Akhir ini. Pembahasan pada Sub bab ini meliputi Rangkaian *Power Supply*, Rangkaian *Pre-Amp*, Rangkaian *Mixer*, Rangkaian MAXIM5406 dengan *Push Button*, *Shield* untuk MAXIM5406, Rangkaian MAXIM5406 dengan *Transistor Optocoupler* yang dikerjakan oleh Nafiqul Ihsan. Konfigurasi Arduino Mega 2560 dengan MAXIM5406 dan Konfigurasi Arduino Mega 2560 dengan *Bluetooth HC-05* dikerjakan oleh Dyah Ayuningtyas.

3.3.1 Rangkaian *Power Supply*

Power Supply adalah perangkat keras yang mampu menyuplai tenaga atau tegangan listrik secara langsung dari sumber tegangan listrik ke tegangan listrik yang lainnya. Rangkaian *Power Supply* dikerjakan oleh Nafiqul Ihsan. *Power supply* biasanya digunakan untuk perangkat elektronika sebagai penghantar tegangan listrik secara langsung kepada

komponen-komponen atau perangkat keras lainnya yang ada di rangkaian tersebut, seperti *hardisk*, kipas, *motherboard* dan lain sebagainya. Pada Gambar 3.7 berikut merupakan rangkaian *power supply*.



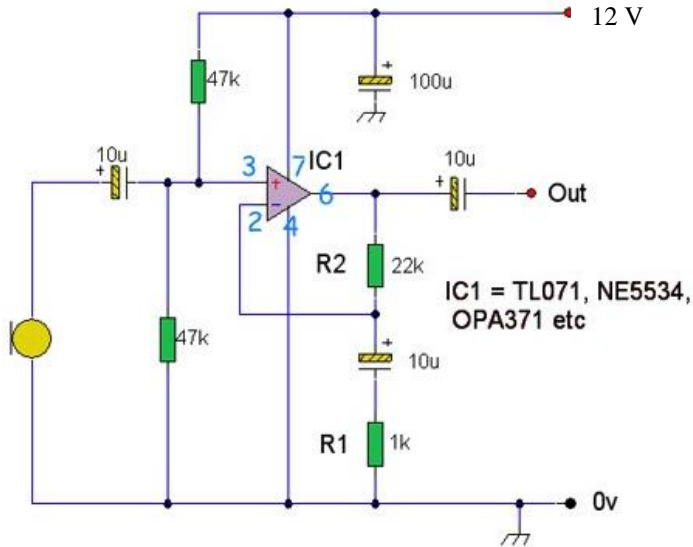
Gambar 3.7 Rangkaian *Power Supply*

Power supply memiliki *input* dari tegangan *Alternating Current* (AC) dan mengubahnya menjadi tegangan *Direct Current* (DC) lalu menyalurkannya ke berbagai perangkat elektronika.

3.3.2 Rangkaian *Pre-Amplifier*

Rangkaian *Pre-Amplifier* menerima sinyal audio dari *mic* kemudian sinyal *mic* akan dikuatkan melalui IC TL081CN yang digunakan sebagai penguatnya. Rangkaian *Pre-Amplifier* dikerjakan oleh Nafiqul Ihsan. Tegangan oleh sinyal; yang dikeluarkan oleh *mic* hanya sebesar 0,1 mV sehingga tidak memungkinkan untuk langsung diterima oleh rangkaian *tone control*. Catu daya ideal untuk rangkaian ini adalah sebesar 12 V. Untuk mengetahui seberapa besar penguatan dari rangkaian *Pre-Amp* ini, kita dapat melihatnya melalui perbandingan dari resistor yang terdapat pada rangkaian khususnya R1 dan R2. Dari Gambar 3.8 terlihat perbandingan R1 dan R2 adalah sebesar 1 K : 22 K. Hal itu berarti penguatan yang dihasilkan oleh rangkaian *Pre-Amp* ini menggunakan IC TL081CN adalah sekitar 22kali dari tegangan masukan dari *input Pre-Amp*. Hal ini dirasa cukup untuk diteruskan masuk ke rangkaian *tone control*.

Pre-amplifier mikrofon berkualitas tinggi menggunakan catu daya tunggal, cocok untuk mikrofon dinamis atau *electric*. *Op-Amp* yang digunakan mempunyai kebisingan rendah seperti TL081CN NE5534, TL071, dan OPA 371.



Gambar 3.8 Rangkaian *Pre-Amplifier*

Didalam IC TL081CN hanya terdapat satu rangkaian *operational amplifier* sehingga untuk satu rangkain *Pre-Amp* dapat dipastikan harus menggunakan 1 IC jenis ini. Untuk mengetahui seberapa penguatan yang dapat dihasilkan, dapat dilihat melalui perbandingan terhadap R1 dan R2 yang terlihat pada Gambar 3.8.

Tidak ada yang special. Desainnya adalah *non-inverting* standar, masukannya diterapkan pada *input non-inverting* yang terdapat pada pin nomor 3. Impedansi masukan adalah 23,5 K, *gain* tegangan keseluruhan ditentukan oleh R2 dan R1 sesuai dengan rumus berikut: [1]

$$V_{out} = (R2/R1) + 1 \quad \dots(3.1)$$

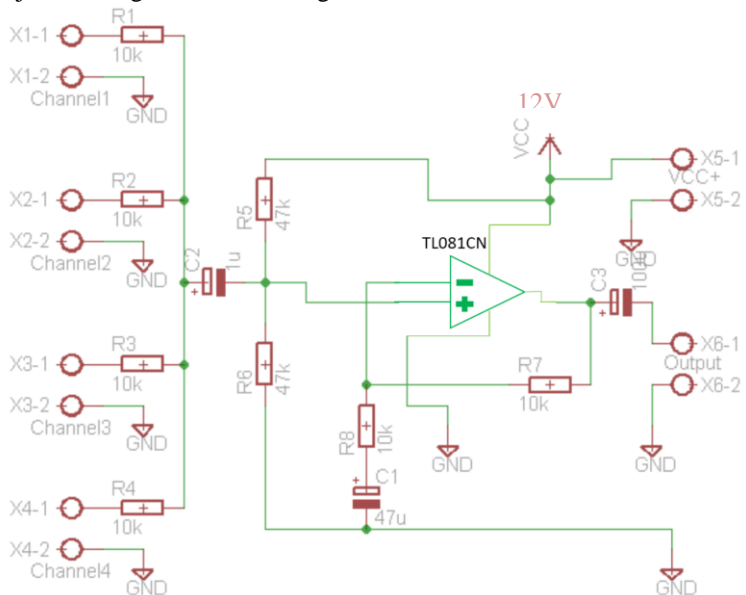
Selain itu suatu rangkaian *Pre-Amp* dapat dihitung penguatannya dalam ukuran desibel(dB). Ada beberapa parameter penguatan dalam satuan dB, yaitu penguatan arus, penguatan daya, dan penguatan tegangan. Untuk penguatan tegangan berikut persamaannya: [1]

$$Penguatan\ Tegangan\ (dB) = 20 \log_{10} (V_{out} / V_{in}) \quad \dots(3.2)$$

3.3.3 Rangkaian Mixer

Pada rangkaian *mixer* sebenarnya hampir mirip dengan rangkaian *Pre-Amp*, namun letak perbedaannya terdapat pada fungsi dari rangkaianannya, yaitu menyatukan semua *input* dari rangkaian *tone control*. Setelah dijadikan satu pada rangkaian *mixer* maka akan dikuatkan secara bersamaan oleh *amplifier*. Dan yang membedakan lagi dari rangkaian *mixer* adalah perbandingan dari R7 dan R8 dibuat memiliki perbandingan yang bernilai 1 : 1. Yang bertanggungjawab untuk pembuatan Rangkaian *Mixer* adalah Nafiqul Ihsan.

Mencampur bermacam sumber sinyal memberikan efek yang indah dan menyenangkan. Secara prinsip mencampur dua atau lebih sumber sinyal sederhananya menghubungkan sumber-sumber sinyal tadi secara paralel seperti yang telah dijelaskan pada Sub bab 2.6. Gambar 3.9 ditunjukkan rangkaian *Mixer* dengan IC TL081CN.

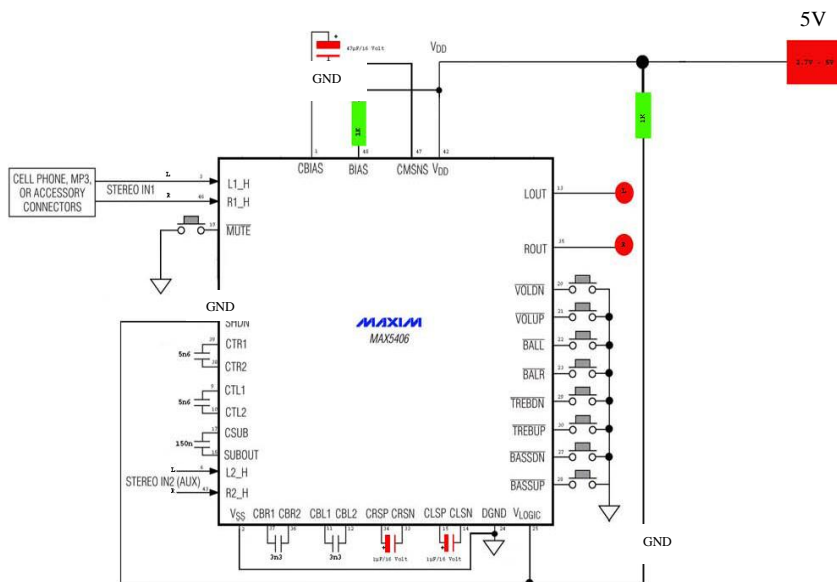


Gambar 3.9 Rangkaian *Mixer*

3.3.4 Rangkaian MAXIM 5406 dengan *Push Button*

Pada awalnya MAXIM5406 ini merupakan IC *tone control* yang dikontrol secara manual menggunakan *push button* seperti yang ditampilkan pada Gambar 3.10, namun sesuai judul Tugas Akhir ini,

maka dirubah agar MAXIM5406 *tone control* ini agar dapat dikontrol secara digital menggunakan *smartphone android* yang berbasis *Iot*. Pada Tugas Akhir ini dilakukan modifikasi dengan mengganti *push button* manual dengan *transistor optocoupler 4n25* agar dapat dikontrol secara digital. Yang bertanggungjawab dalam pembuatan Rangkaian MAXIM5406 dengan *Push Button* adalah Nafiqul Ihsan. Serta dapat memisahkan tegangan dalam sistem dan tegangan dari luar sistem.



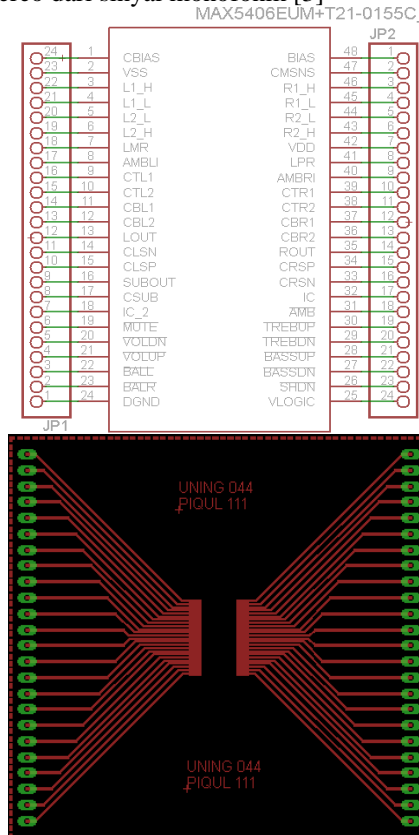
Gambar 3.10 Rangkaian MAXIM5406 dengan *Push Button*

3.3.5 Shield untuk MAXIM 5406

Pembuatan *shield* ini bermaksud untuk mempermudah pekerjaan ,dikarenakan IC yang digunakan berukuran smd sehingga rawan akan problem jika tidak menggunakan *shield*. Yang bertanggungjawab dalam pembuatan *Shield* untuk MAXIM5406 adalah Nafiqul Ihsan. Dengan adanya *shield* ini maka terdapat rangkaian atas dan rangkaian bawah seperti yang ditunjukkan pada Gambar 3.11. Jika yang bermasalah adalah rangkaian yang bawah maka tinggal melepas *shield* atas.

MAX5406 juga dilengkapi dengan kontrol suasana untuk meningkatkan pemisahan output saluran kiri dan kanan untuk

headphone dan sistem *speaker desktop*, dan *pseudostereo*. Fitur yang mendekati suara stereo dari sinyal monofonik [3]



Gambar 3.11 *Shield* untuk MAXIM5406

3.3.6 Rangkaian MAXIM5406 dengan *Transistor Optocoupler 4n25*

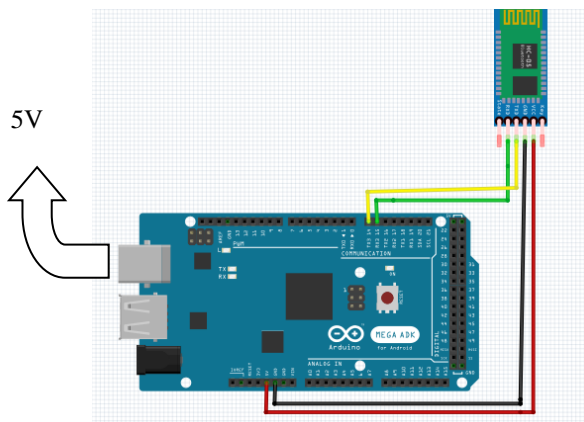
Pada Gambar 3.12 terlihat rangkaian MAXIM5406 menggunakan *transistor optocoupler 4n25* sebagai pemisah tegangan dalam dan tegangan luar serta agar dapat dikontrol secara digital menggunakan Arduino Mega2560. Yang bertanggungjawab dalam pembuatan Rangkaian MAXIM5406 dengan *Transistor Optocoupler 4n25* adalah Nafiqul Ihsan. Struktur dari *transistor optocoupler* sendiri yaitu terdapat led dioda di dalamnya, maka dari itu sebelum masuk ke *anode* dioda

Tabel 3.1 Daftar *Wiring* Arduino Mega 2560 dengan MAXIM5406

Pin MAXIM5406	Pin Arduino Channel1	Pin Arduino Channel2	Pin Arduino Channel3	Pin Arduino Channel4
<i>TREBUP</i>	37	48	34	35
<i>TREBDN</i>	39	46	32	33
<i>BASSUP</i>	43	42	28	29
<i>BASSDN</i>	45	40	26	27
<i>VOLUP</i>	47	38	24	25
<i>VOLDN</i>	49	36	22	23
<i>MUTE</i>	41	44	30	31

3.3.8 Konfigurasi Arduino Mega 2560 dengan *Bluetooth*

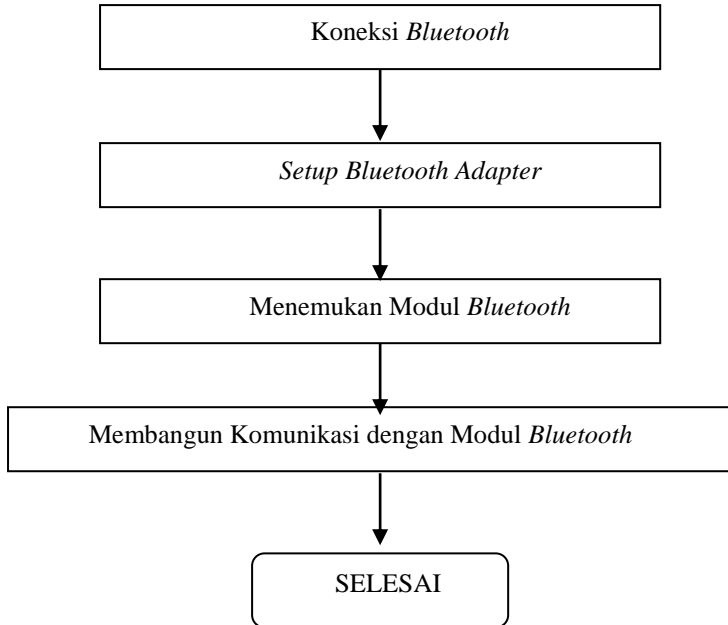
Konfigurasi Arduino Mega 2560 dengan *Bluetooth* tidaklah rumit. Pada *Bluetooth* terdapat 6 pin namun yang digunakan hanya 4 yaitu TXD, RXD, GND, dan VCC. TXD disambungkan pada pin RXD pada Arduino, sedangkan RXD disambungkan pada TXD pada pin Arduino. Pin GND disambungkan pada pin GND pada Arduino dan VCC disambungkan pada pin 5 V seperti yang ditunjukkan pada Gambar 3.13. Yang bertanggungjawab adalah Dyah Ayuningtyas.



Gambar 3.13 Konfigurasi Arduino Mega 2560 dengan *Bluetooth* HC-05

Untuk membuat program yang akan dijalankan di *smartphone android* yang menggunakan koneksi *Bluetooth*, dibawah ini adalah langkah-langkah membuat program *android* yang menggunakan fitur

komunikasi melalui koneksi *Bluetooth* seperti yang ditunjukkan pada diagram alur Gambar 3.14.



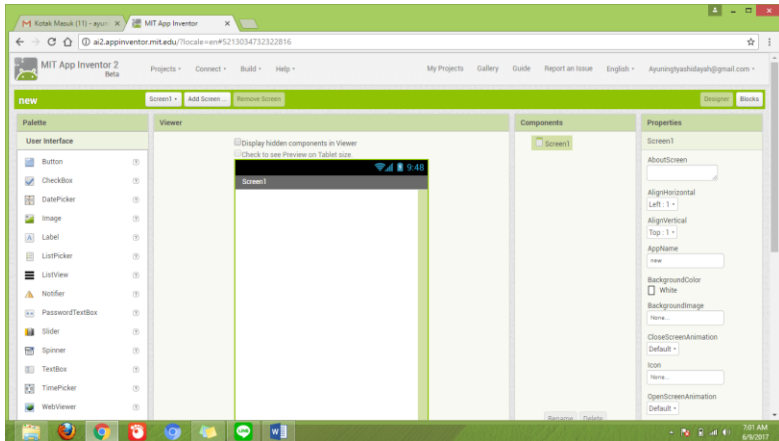
Gambar 3.14 Diagram Alur Proses Koneksi *Bluetooth*

3.4 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak dibahas tentang *software* yang digunakan dalam Tugas Akhir ini. Pembahasan pada perancangan perangkat lunak ini meliputi perancangan aplikasi *android* dan perancangan *software* Arduino. Yang bertanggungjawab adalah Dyah Ayuningtyas.

3.4.1 Perancangan Aplikasi *Android*

Aplikasi yang dibuat akan digunakan sebagai *remote* untuk mengatur *tone control* pada *mixer* audio. Perancangan aplikasi ini menggunakan *software MIT App Inventor 2* yang diakses secara *online* dan masing-masing pengguna diharuskan memiliki akun terlebih dahulu. Yang bertanggungjawab adalah Dyah Ayuningtyas. Setelah berhasil masuk pada akun, akan tampil halaman utama *MIT App Inventor 2*. Untuk membuat desain aplikasi baru yaitu dengan klik *Project – Start New Project*. Maka akan tampil *design view* seperti pada Gambar 3.15.



Gambar 3.15 Tampilan *Design View*

Pada *Design View* terdapat bermacam-macam menu pilihan, antara lain menu *Pallete*, *Viewer*, *Components*, dan *Properties*. Pada Menu *Pallete* memiliki sub menu yang berisi *icon-icon* yang digunakan untuk *interface* maupun pendukung lainnya seperti *Bluetooth*, *Timer*, dan lainnya. Pada menu *Viewer* merupakan tempat merancang *user interface* yang dibuat, yang akan tampil pada *android*. Pada menu *Components* akan menampilkan daftar *icon* apa saja yang telah digunakan atau dipilih. Pada menu *Properties* terdapat pengaturan-pengaturan yang menunjang setiap *icon* seperti nama, ukuran tulisan, warna yang diinginkan, dan lainnya.

Setelah merancang *user interface*, akan tampil seperti Gambar 3.16 yaitu menu utama pada aplikasi yang muncul di *android*. Pada menu

utama terdapat *icon-icon* yang memiliki fungsi masing-masing, fungsi-fungsi *icon* yang digunakan akan dijelaskan pada Tabel 3.2.




Gambar 3.16 Tampilan Menu Utama Aplikasi



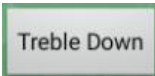
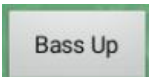
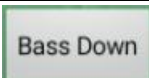

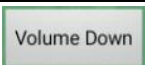
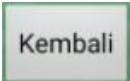

Setelah menu utama selanjutnya menu pada *channel*. Menu pada *channel* akan muncul ketika tombol *Channel 1* ditekan. Menu *Channel 1*, *Channel 2*, *Channel 3*, dan *Channel 4* adalah sama seperti yang ditunjukkan pada Gambar 3.17. Pada menu *Channel* inilah *sound engineer/operator* dapat mengatur *tone control* sesuai dengan keinginannya masing-masing. Pada Menu *Channel* terdapat tombol, *Treble Up*, *Treble Down*, *Bass Up*, *Bass Down*, *Volume Up*, *Volume Down*, serta *Mute*. Yang mana masing-masing penjelasan tombol dijelaskan pada Tabel 3.2. Setelah *Bluetooth* dikoneksikan dengan cara meng-klik gambar *Bluetooth* pada menu utama, selanjutnya masuk pada menu *Channel* yang ingin diatur suaranya. Ketika tombol ditekan, maka *Bluetooth* mengirim data kepada rangkaian *tone control* digital untuk melakukan pekerjaannya sesuai dengan data yang dikirimkan.



Gambar 3.17 Tampilan Menu pada *Channel*

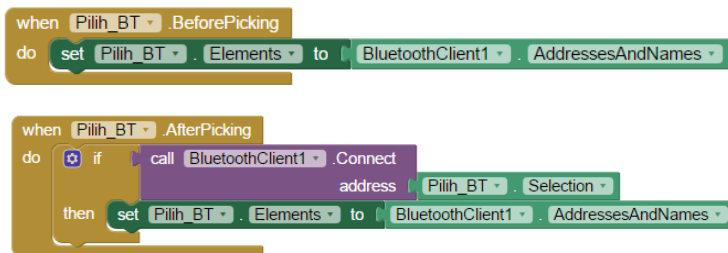
Tabel 3.2 Penjelasan *Icon* pada Aplikasi *Android*

<i>Icon</i>	<i>Penjelasan</i>
Mixer Audio	<i>Icon</i> Label <i>Mixer</i> Audio yang menampilkan tulisan / keterangan yang diperlukan
	<i>Icon</i> <i>List Picker Bluetooth</i> yang berfungsi untuk memilih dan mengkoneksikan ke <i>Bluetooth</i>
Tidak Terhubung	<i>Icon</i> Label Tidak Terhubung / Terhubung yang menampilkan status koneksi <i>Bluetooth</i>

Icon	Penjelasan
	<i>Icon Button Channel</i> untuk memilih <i>Channel</i> yang digunakan untuk mengatur masing-masing <i>Channel</i>
	<i>Icon Button Treble Up</i> yang berfungsi untuk menaikkan <i>Treble</i>
	<i>Icon Button Treble Down</i> yang berfungsi untuk menurunkan <i>Treble</i>
	<i>Icon Button Bass Up</i> yang berfungsi untuk menaikkan <i>Bass</i>
	<i>Icon Button Bass Down</i> yang berfungsi untuk menurunkan <i>Bass</i>
	<i>Icon Button Volume Up</i> yang berfungsi untuk menaikkan <i>Volume</i>
	<i>Icon Button Volume Down</i> yang berfungsi untuk menurunkan <i>Volume</i>
	<i>Icon Button Kembali</i> yang berfungsi untuk kembali ke menu utama
	<i>Icon Button Mute</i> yang berfungsi untuk mem-pause <i>Channel</i>

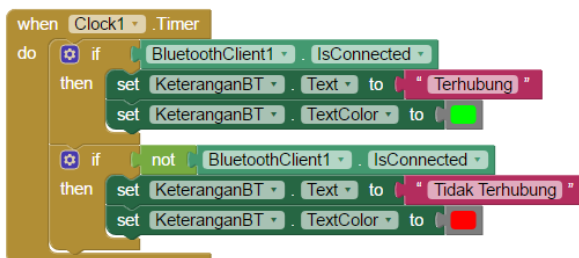
Setelah perancangan *user interface* pada *design view* selesai selanjutnya menyusun *code block*. Untuk masuk ke *block editor* klik *Blocks* pada bagian kanan atas *design view*.

Penyusunan *block* digunakan untuk mengatur jalannya program sesuai dengan yang diinginkan dan diharapkan oleh pengguna. *Block* yang dapat digunakan disesuaikan dengan *icon* yang terdapat pada menu *components* atau *icon* yang dipilih. Sehingga *icon* yang tidak dipilih tidak akan muncul pada pilihan *block* di *block editor*. Dengan demikian kemungkinan kesalahan memilih *block* akan lebih kecil. Selain itu *block* akan secara otomatis menolak untuk bergabung apabila *block* tersebut tidak sesuai dengan kriteria *block* utama tadi. Berikut penjelasan masing-masing *block* yang sudah disusun sesuai jalannya program.



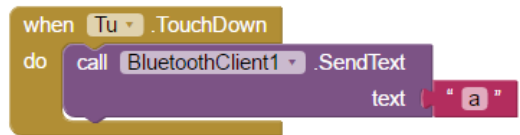
Gambar 3.18 *Block Program Konektivitas Bluetooth*

Gambar 3.18 menunjukkan *block* program konektivitas *Bluetooth*. Dengan demikian, ketika *icon* Pilih_BT ditekan maka akan tampil daftar alamat *Bluetooth* yang akan dipasangkan, kemudian pilih alamat *Bluetooth* yang sesuai untuk dihubungkan dengan perangkat keras, kemudian program akan memanggil *Bluetooth* tersebut untuk dipasangkan.



Gambar 3.19 *Block Program Status Koneksi Bluetooth*

Pada Gambar 3.19 merupakan *block* program status koneksi *Bluetooth* setelah dipasangkan sebelumnya. Ketika *Bluetooth* telah terhubung, maka akan menampilkan teks “Terhubung” dengan teks berwarna hijau. Apabila koneksi *Bluetooth* tidak terkoneksi, maka akan tampil teks “Tidak Terhubung” dengan teks berwarna merah.



Gambar 3.20 Block Program Tombol Tu Ketika di *TouchDown*

Gambar 3.20 menunjukkan *block* program ketika tombol Tu ditekan. Tombol Tu merupakan tombol pengiriman data “a” untuk mengatur *tone control* yang dikirim ke Arduino Mega 2560. Tabel 3.3 menunjukkan data yang dikirim oleh tombol-tombol pada menu *Channel*.

Tabel 3.3 Data yang Dikirim ke Arduino Mega 2560

Nama Tombol	Channel 1	Channel 2	Channel 3	Channel 4
Tu(<i>Trebleup</i>)	A	h	o	V
Td(<i>Trebel</i> dn)	B	i	p	W
Bu(<i>Bassup</i>)	C	j	q	X
Bd(<i>Bass</i> dn)	D	k	r	Y
Vu(<i>Volumeup</i>)	G	l	s	Z
Vd(<i>Volume</i> dn)	F	m	t	A
Mute	G	n	u	B

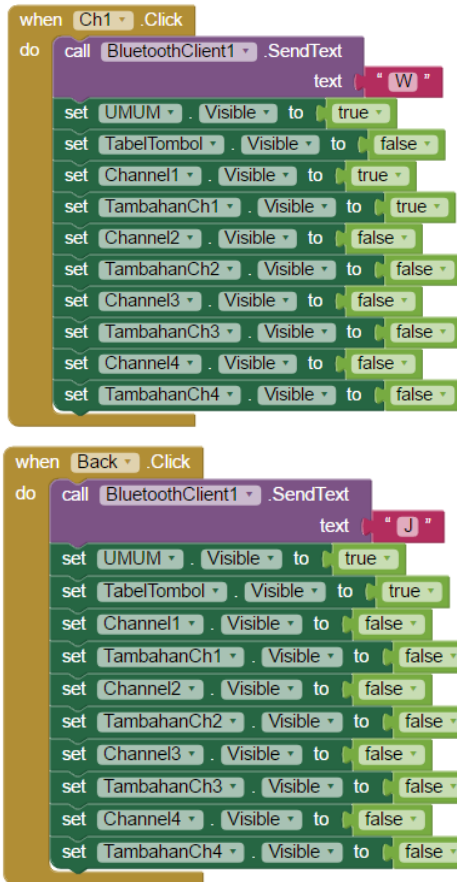
Setelah Arduino Mega 2560 mendapatkan perintah data “a”, maka selanjutnya Arduino Mega 2560 mengirim perintah tersebut kepada MAXIM5406 melalui pin 27 seperti yang disebutkan pada Tabel 3.3 dan mengaktifkan *Optocoupler Transistor* menjadi aktif *High* satu kali. Begitu juga dengan tombol pada pilihan yang lainnya.



Gambar 3.21 Block Program Ketika Tombol Tu di *Touch Up*

Pada Gambar 3.21 menunjukkan keadaan ketika tombol Tu di *Touch Up* atau dilepas/tidak ditekan. Terlebih dahulu kondisi awal diinisialisasi terlebih dahulu yaitu dalam kondisi “*off*”. Sehingga ketika tombol Tu di lepas/tidak ditekan, maka *Bluetooth* akan memberi perintah untuk “*off*” atau tidak mengirim data “a” lagi.

Gambar 3.22 menunjukkan block program untuk menginisialisasi layar pada menu utama, pada aplikasi yang digunakan Tugas Akhir ini menggunakan *multiscreen* untuk mengganti *layout* yang muncul ketika menekan tombol tertentu. Saat *layout* Ch1 ditekan maka *layout* UMUM, *Channel1*, *TambahanCh1* akan tampil, dan *layout* TabelTombol, *Channel2*, *TambahanCh2*, *Channel3*, *TambahanCh3*, *Channel4*, *TambahanCh4* tidak muncul. Muncul dan tidaknya suatu *layout* diatur dengan logika “*true*” dan “*false*”. Begitu juga untuk Ch2, Ch3, dan Ch4 yang dijelaskan pada lampiran.



Gambar 3.22 Block Program Inialisasi *Layout*

3.4.2 Perancangan *Software Arduino*

Dalam Pembuatan perangkat lunak pada Tugas Akhir ini, *software* Arduino juga diperlukan. Dengan menggunakan program yang sederhana, program ini mampu membuat perangkat keras bekerja. Berikut penjelasan masing-masing program yang digunakan.

```
char val = 0;
```

Gambar 3.23 Deklarasi Nilai *Char*

Tanda *char* mendefinisikan karakter, dan menyediakan operasi dasar dan predikat pada nilai tipe itu. Ada pemesanan linear yang didefinisikan pada karakter. Program untuk mendeklarasikan *char* ditunjukkan pada Gambar 3.23.

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode (37,OUTPUT);  
    pinMode (39,OUTPUT);  
    pinMode (43,OUTPUT);  
}
```

Gambar 3.24 Fungsi *Setup*

Fungsi *setup()* pada Gambar 3.24 dipanggil ketika sketsa dimulai. Struktur ini berguna untuk menginisialisasi variabel, mode pin, memulai menggunakan *library*, dan lain lain. Fungsi pengaturan hanya akan berjalan sekali, yaitu setiap *powerup* atau *restart board* Arduino.

Pada program ini, pin yang digunakan ditunjukkan pada Tabel 3.1 pada bab sebelumnya. Pin ini diinisialisasi untuk menerima data yang dikirim oleh *android*.

```
Serial.begin(9600);
```

Gambar 3.25 Kecepatan Transmisi Data

Pada program digunakan perintah *Serial.begin(9600)* Gambar 3.25 maksudnya adalah untuk mengatur kecepatan data dalam bit per detik (*baud*) untuk transmisi data serial. Tarif baud yang digunakan untuk berkomunikasi dengan komputer pada program ini adalah 9600.

```
void loop() {  
    // put your main code here, to run repeatedly:  
    if(Serial.available ()>0) {  
        val = Serial.read();  
    }  
}
```

Gambar 3.26 Eksekusi Program

Program pada *void loop ()* digunakan untuk mengontrol *board* Arduino baik membaca *input* atau merubah *output* dengan cara melaksanakan / mengeksekusi perintah program yang telah dibuat. Telah ditunjukkan pada Gambar 3.26 fungsi ini akan terus berjalan sampai *reset*. Fungsi *Serial.available ()* digunakan untuk mengetes apakah ada *input* data dari *hardware* yang disambungkan ke *serial port*, misalnya dari PC. Fungsi ini akan menghasilkan 1 apabila ada masukan, dan 0 apabila tidak ada masukan.

```
if (val == 'a'){  
    digitalWrite(37,HIGH);  
    delayMicroseconds(1000);  
    digitalWrite(37,LOW);  
    Serial.print("a");  
}
```

Gambar 3.27 Proses Data

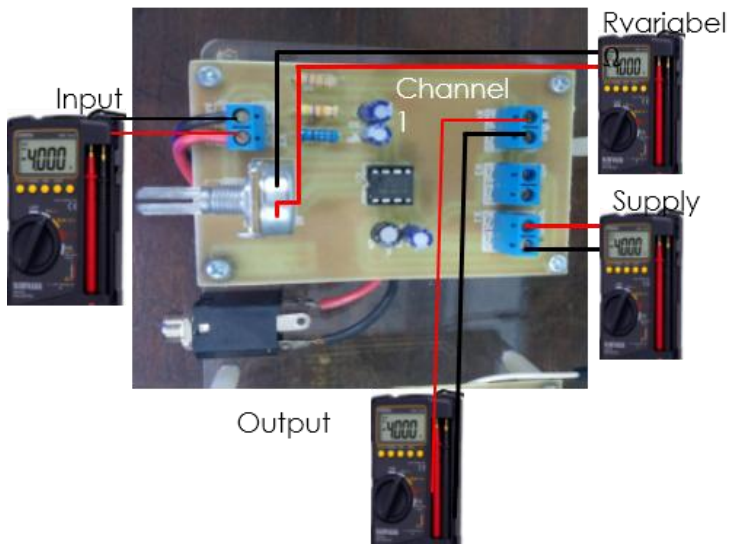
Program yang dijalankan pada Gambar 3.27 ditunjukkan bahwa ketika data yang diterima adalah 'a', maka akan menyalakan atau memberikan tegangan pada pin 37 dengan *delayMicroseconds(1000)* setelah itu off. Kemudian *Serial.print ("a")* merupakan perintah untuk menampilkan "a" ketika kondisi *High*.

BAB IV PENGUJIAN DAN ANALISA DATA

Bab ini akan dijelaskan mengenai uji coba pada masing masing komponen dan juga data dari komponen. Pada bab ini Nafiqul Ihsan bertanggung jawab pada pegujian yang berhubungan dengan rangkaian *Pre-amp*, *Tone Control*, dan *Mixer* yang terdapat pada poin 4.1 4.1, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, dan 4.12. Begitu juga dengan Dyah Ayuningtyas bertanggungjawab pada pengujian yang berhubungan dengan perancangan *software* dan pengiriman data *android* yang terdapat pada poin 4.10, 4.11, 4.12.

4.1 Pengujian *Pre-Amp Channel 1*

Pre-Amp channel 1 menggunakan *micrhopone* jenis dinamik dengan frekuensi 100-12.000 Hz . Pada Gambar 4.1 ditunjukkan tampilan *Pre-Amp channel 1*.



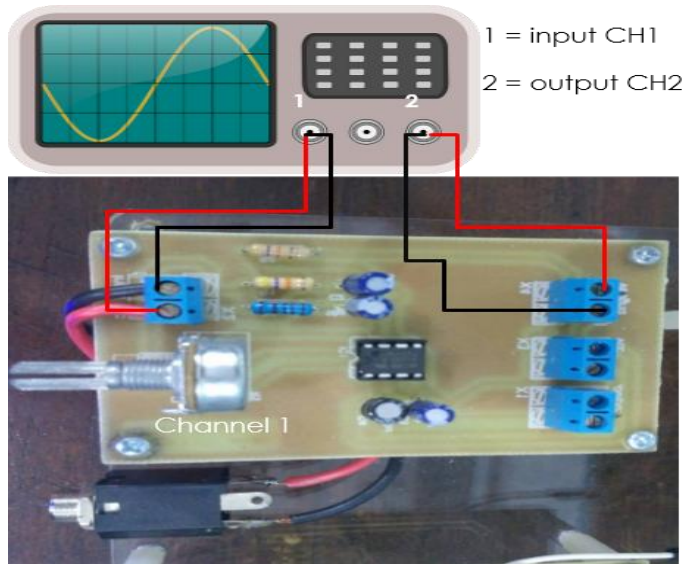
Gambar 4.1 Pengujian *Pre-Amp Channel 1*

Pada rangkaian *Pre-Amp*, menggunakan penguatan *non inverting* dengan IC TL081CN dimana untuk R potensio sebesar 50 K Ω . Dan untuk perbandingan antara R1 dan R2 sebagai pembanding seberapa

besar penguatan yang dilakukan oleh rangkaian adalah sebesar 1:50 seperti yang disebutkan pada Persamaan 3.1 dan untuk *supply* untuk *Pre-Amp* sendiri sebesar 12,36 Vdc. Tabel spesifikasi dari *pre-amp channel 1* terlihat pada Tabel 4.1.

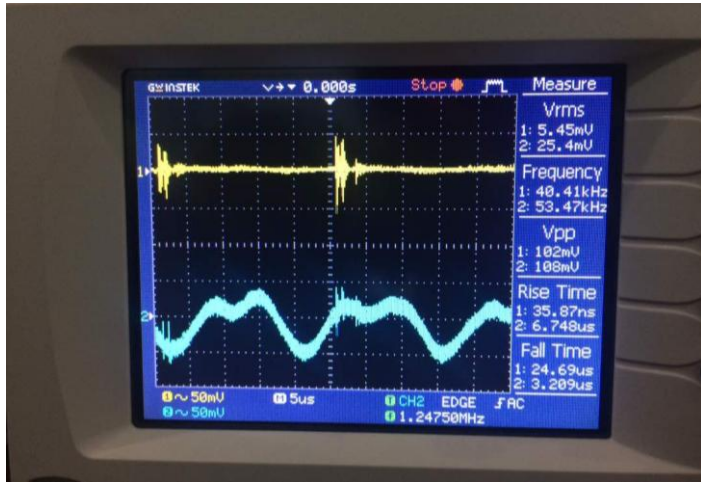
Tabel 4.1 Spesifikasi Rangkaian *Pre-Amp Channel 1*

Supply <i>Pre-Amp</i>	R Porentsio	Vin <i>Microphone</i>	Vout <i>Pre-Amp</i>	Alat <i>Input</i>
12,36 V	50,1 K Ω	11,4 mV	2,5 V	Mic Dinamis 10- 12.000 Hz



Gambar 4.2 Pengujian *Input* dan *Output Pre-Amp Channel 1*

Pengambilan data menggunakan *oscilloscope* untuk mengetahui gelombang *input* dan *output* ditampilkan seperti pada Gambar 4.2



Gambar 4.3 Gelombang *Input* dan *Output Pre-Amp channel 1*

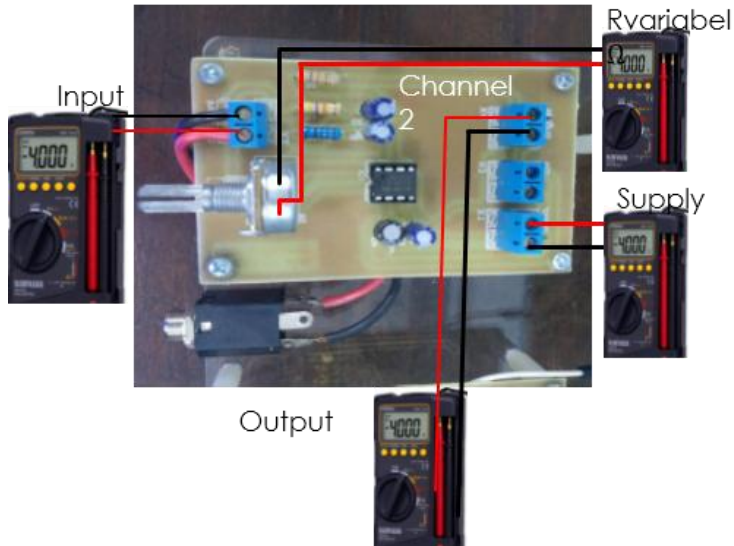
Pengambilan data menggunakan *oscilloscope* terlihat pada Gambar 4.3 dimana terjadi penguatan sinyal yang dilakukan oleh *Pre-Amp* melalui IC TL081CN dengan perbandingan antara R1 dan R2 adalah sebesar 1:50. Selain itu terdapat pula pengujian penguatan dalam satuan dB dengan menggunakan Persamaan 3.2 untuk penguatan dengan karakteristik tegangan :

$$\begin{aligned}
 &= 20 \text{ Log } (2,5\text{V}/0,011\text{V}) \\
 &= 20 \text{ Log } 227,27 \\
 &= 20 (2,35) \\
 &= 47,13 \text{ dB}
 \end{aligned}$$

Dari perhitungan diatas terlihat penguatan yang dilakukan oleh rangkaian *Pre-Amp Channel 1* sebesar 47,13 dB.

4.2 Pengujian *Pre-Amp Channel 2*

Pada *Pre-Amp channel* dua memiliki spesifikasi yang sama dengan *channel 1* yaitu menggunakan IC TL081CN dimana *input* yang digunakan adalah *microphone* dinamis dengan *range* frekwensi 50 – 15.000 Hz. *Output* sinyal audio yang dihasilkan oleh *microphone* terhitung kecil berkisar 11 – 50 mV, kemudian setelah melalui *Pre-Amp* maka sinyal *output* audio menjadi lebih besar berkisar 2,5 -3 V. Pada Gambar 4.4 merupakan tampilan dari rangkaian *Pre-Amp Channel 2*.



Gambar 4.4 Pengujian *Pre-Amp Channel 2*

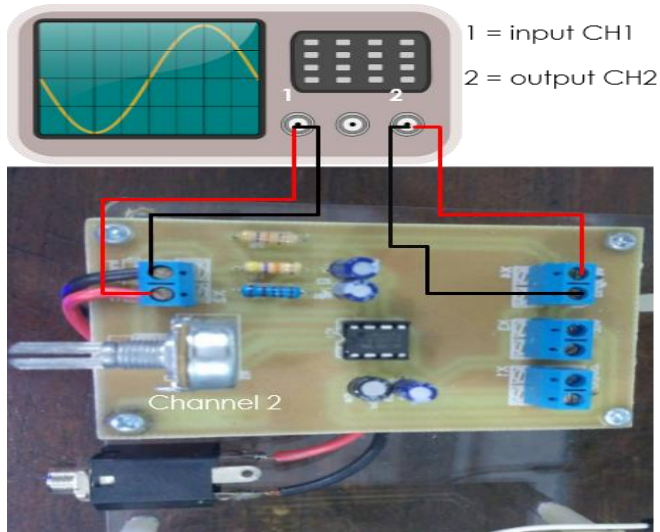
Pada rangkaian *Pre-Amp channel 2* menggunakan R potensio sebesar 50K Ω dengan penguatan yang disebutkan pada Persamaan 3.1 yang menggunakan IC TL081CN sebagai penguat. Pada Tabel 4.2 menyajikan spesifikasi yang di hasilkan oleh rangkaian *Pre-Amp channel 2*.

Tabel 4.2 Spesifikasi Rangkaian *Pre-Amp Channel 2*

Supply <i>Pre-Amp</i>	R Potensio	V_{in} <i>Microphone</i>	V_{out} <i>Pre-Amp</i>	Alat <i>Input</i>
12,36 V	49,5 K Ω	10,6 mV	2,4 V	Mic Dinamis 50-15.000 Hz

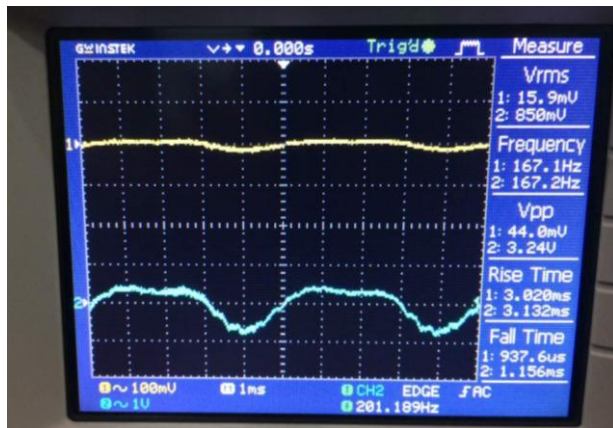
Berdasarkan Tabel 4.2 disajikan data *input* serta *output* yang dihasilkan oleh *Pre-Amp channel 2* yaitu terjadi penguatan yang semula *input* dari *microphone* dengan frekuensi 50-15.000 Hz sebesar 10,6 mV sehingga untuk keluarannya menjadi sekitar 2,5-3 V. Untuk

pengambilan data gelombang menggunakan *oscilloscope* yang ditampilkan pada Gambar 4.6.



Gambar 4.5 Pengujian *Input* dan *Output Pre-Amp Channel 2*

Pada Gambar 4.5 terlihat pengujian *pre-amp channel 2* dimana untuk mengetahui penguatan *input* dan *output*.



Gambar 4.6 Gelombang *Input* dan *Output Pre-Amp Channel 2*

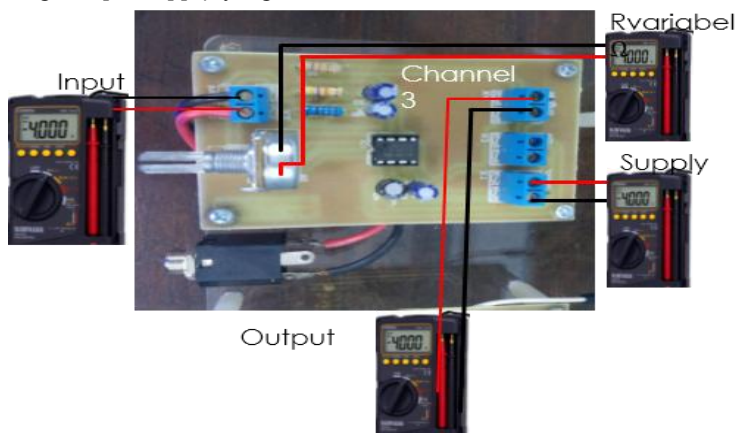
Berdasarkan pengujian data menggunakan *oscilloscope* maka terlihat perubahan *amplitude* pada Gambar 4.6 yaitu dimana awal *input* berkisar 10,6 mV setelah dikuatkan maka menjadi 2,4 V. Untuk mengetahui seberapa besar penguatan yang dilakukan oleh rangkaian, maka kita menggunakan Persamaan 3.2 dengan penguatan tegangan dengan persamaan berikut :

$$\begin{aligned} &= 20 \log (2,4\text{V}/0,01\text{V}) \\ &= 20 \log 240 \\ &= 20 (2,38) \\ &= 47,6 \text{ dB} \end{aligned}$$

Dari perhitungan terlihat penguatan yang dilakukan oleh *channel 2* yaitu yang *input* tegangan awal sebesar 10,6 mV dan keluarannya menjadi 2,4 V. Setelah di masukkan dalam persamaan terlihat terjadi penguatan sebesar 47,6 dB.

4.3 Pengujian Pre-Amp Channel 3

Karakteristik pada *Pre-Amp channel 3* sama dengan *channel 1* dan *channel 2* seperti yang terlihat pada Gambar 4.7, yaitu menggunakan R potensio sebesar 50 K Ω yang perbandingan sebesar 1 :50 seperti yang disebutkan pada Persamaan 3.1. Dimana untuk *inputnya* menggunakan gitar listrik yang *inputnya* lebih besar dari *input microphone* yaitu berkisar 170 mV hingga 200 mV dan untuk keluarannya berkisar 2,5 V dengan *input supply* yang masuk sebesar 12,36 V.

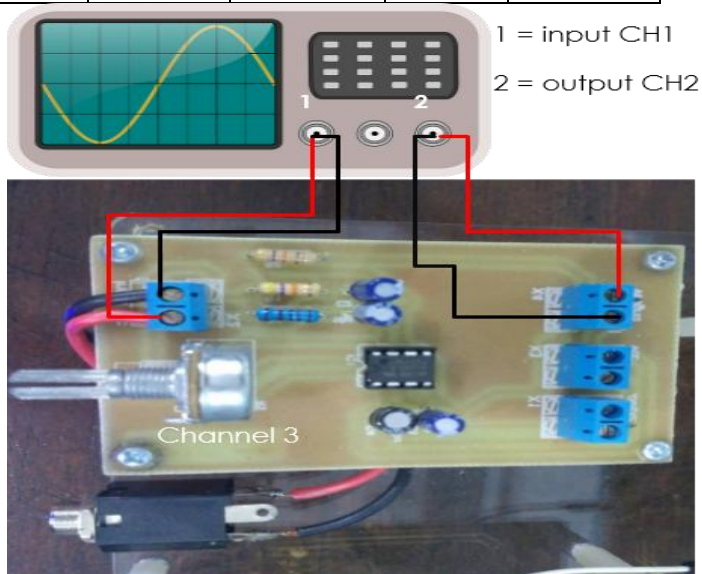


Gambar 4.7 Pengujian *Pre-Amp Channel 3*

Gambar 4.7 menunjukkan tampilan Rangkaian *Pre-Amp* dengan IC TL081CN yang menggunakan instrument gitar listrik sebagai *input*nya dimana sinyal keluaran yang dihasilkan oleh gitar listrik memiliki tegangan yang lebih besar dari *microphone* dinamis yaitu berkisar 100 mV. Tampilan spesifikasi ditampilkan oleh Tabel 4.3.

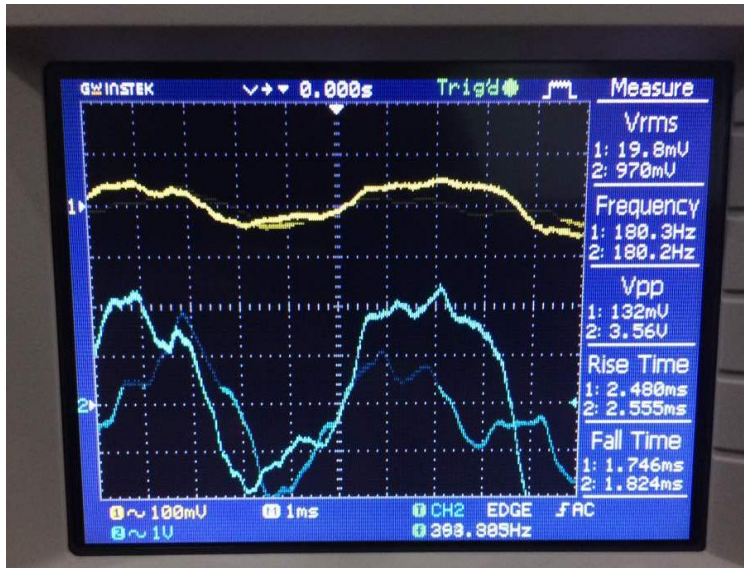
Tabel 4.3 Spesifikasi Rangkaian *Pre-Amp Channel 3*

Supply <i>Pre-Amp</i>	R Potensio	V _{in} <i>Microphone</i>	V _{out} <i>Pre-Amp</i>	Alat <i>Input</i>
12,36 V	50 K Ω	177,64 mV	2,5 V	Gitar Listrik



Gambar 4.8 Pengujian *Input* dan *Output Pre-Amp Channel 3*

Berdasarkan Tabel 4.3 terlihat beberapa spesifikasi baik *input* maupun *output* yang dihasilkan oleh rangkaian *Pre-Amp channel 3* dengan *input supply* 12,36V dengan *input* gitar listrik, dengan konfigurasi seperti pada Gambar 4.8.



Gambar 4.9 Gelombang *Input* dan *Output Pre-Amp Channel 3*

Dari hasil pengujian menggunakan *oscilloscope* pada Gambar 4.9, rangkaian *Pre-Amp channel 3* ini terlihat Vrms yang semua *input* hanya sebesar 19,8 mV kemudian dikuatkan menjadi 970 mV oleh rangkaian *Pre-Amp channel 3* yang menggunakan ICTL081CN dengan penguat inverting. Untuk *input* gitar *electric* maka kita akan menghitung penguatan tegangannya menggunakan Persamaan 3.2:

$$= 20 \log (2,5V/0,177V)$$

$$= 20 \log (14,12)$$

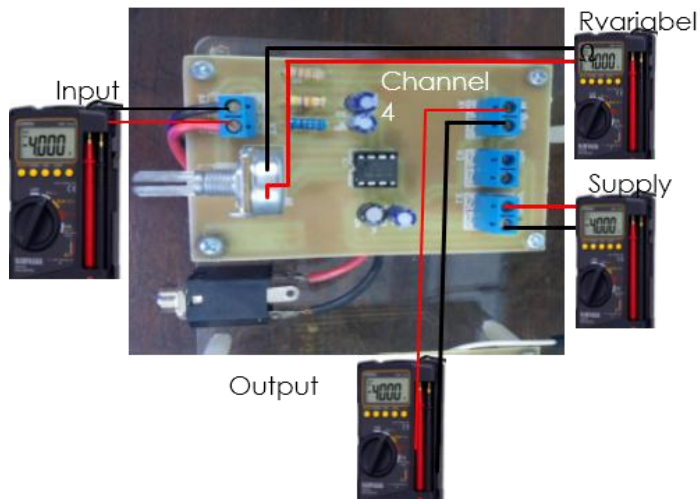
$$= 20 (1,15)$$

$$= 22,9 \text{ dB}$$

Terlihat pada perhitungan diatas, dengan *input* gitar *electric* maka rangkaian melakukan suatu penguatan yang dari *input* sekitar 177,64 mV dengan *output* 2,5 V. Besar penguatan yang dilakukan oleh rangkaian adalah sebesar 22,9 dB. Selain itu juga terlihat dari Gambar 4.9 bahwa terjadi penguatan dilihat dari Vrms.

4.4 Pengujian *Pre-Amp Channel 4*

Rangkaian *Pre-Amp Channel 4* pada Gambar 4.10 menggunakan Resistor Potensio 50 K Ω dengan *Input Keyboard Instrument* dengan *Supply input* sebesar 12,36 V. Untuk sinyal audio yang dihasilkan oleh *instrument keyboard* ini lumayan besar yaitu sekitar 1,8 V namun setelah keluar dari rangkaian *Pre-Amp channel 4* ini tegangannya menjadi 2,5 V. Dengan penguatan 50 kali seperti yang disebutkan pada Persamaan 3.1.

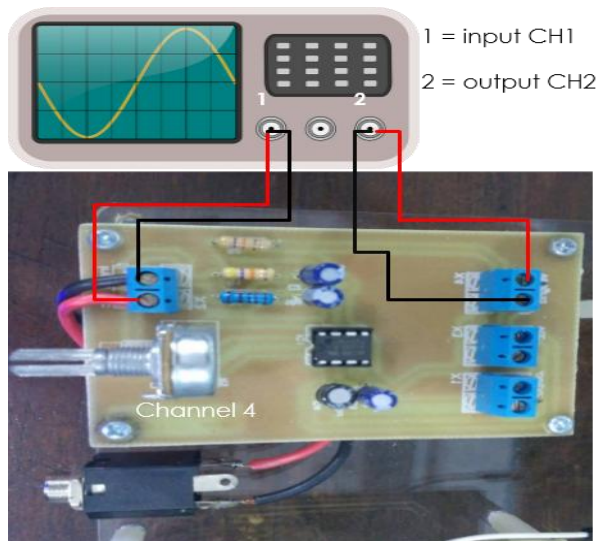


Gambar 4.10 Pengujian *Pre-Amp Channel 4*

Pada Gambar 4.10 terlihat tampilan dari rangkaian *Pre-Amp channel 4* dengan R potensio 50 K Ω dan menggunakan *Keyboard Instrument* sebagai *input*nya. Pada Tabel 4.4 terlihat spesifikasi dari *pre-amp channel 4*.

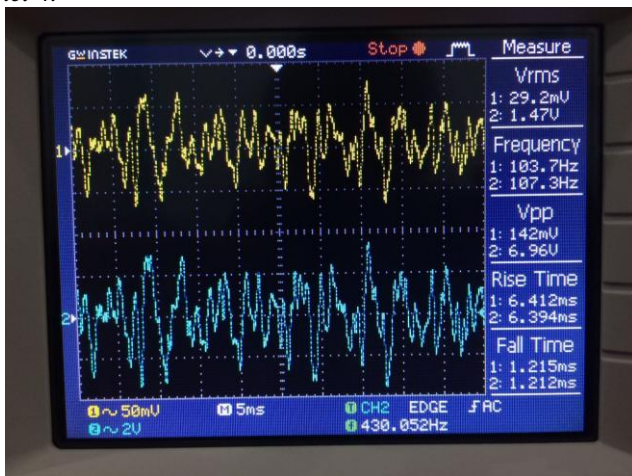
Tabel 4.4 Spesifikasi Rangkaian *Pre-Amp Channel 4*

<i>Supply Pre-Amp</i>	R Potensio	<i>V_{in} Keyboard</i>	<i>V_{out} Pre-Amp</i>	<i>Alat Input</i>
12,36 V	50 K Ω	1,8 V	2,53 V	<i>Keyboard Instrument</i>



Gambar 4.11 Pengujian *Input* dan *Output* Pre-Amp Channel 4

Pada Gambar 4.11 terlihat konfigurasi pengujian sinyal *input* dan *output* untuk melihat penguatan yang dilakukan oleh rangkaian *pre-amp channel 4*.



Gambar 4.12 Gelombang *Input* dan *Output* Pre-Amp Channel 4

Dari Gambar 4.12 terlihat bahwa sinyal yang dihasilkan oleh *instrument keyboard* terjadi penguatan yang dilakukan oleh rangkaian *Pre-Amp channel 4*. Dimana dengan *input* sebesar 29,2 mV dapan dikuatkan menjadi 1,47 V. Untuk perhitungan penguatan yang dilakukan oleh rangkaian menggunakan Persamaan 3.2 sebagai berikut :

$$=20 \log (2,5V/1,8V)$$

$$=20 \log (1.39)$$

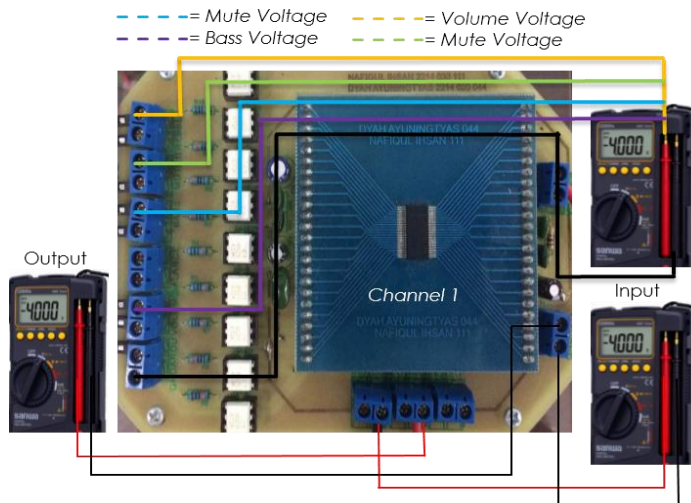
$$=20 (0,14)$$

$$=2,86 \text{ dB}$$

Dari hasil perhitungan terlihat bahwa penguatan yang dilakukan oleh rangkaian *Pre-Amp channel 1* adalah sebesar 2,86 dB terhadap *input keyboard instrument*.

4.5 Pengujian Rangkaian MAXIM 5406 Channel 1

Pengambilan data pada rangkaian *tone control digital* pada Gambar 4.13 menggunakan MAXIM5406 meliputi *input* tegangan yang masuk dan *input* tegangan yang keluar serta bentuk gelombang *input* output. Dimana *input* dari rangkaian *tone control* ini berasal dari *output* dari rangkaian *Pre-Amp channel 1*. Serta mengamati tegangan keluaran dari MAXIM5406 ketika kondisi awal dan kondisi *Volume*, *Bass*, *Treble* dalam keadaan maksimum.

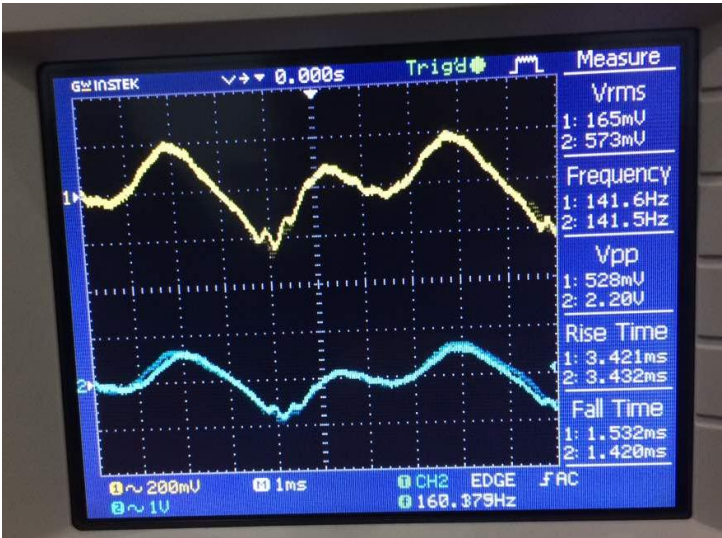


Gambar 4.13 *Tone Control Digital MAXIM5406 Channel 1*

Gambar 4.13 menampilkan tampilan rangkaian *tone control digital* MAXIM5406 *channel 1* yang mendapat *input* dari *Power Supply* sebesar 5,06 V. Pada sisi kirinya terdapat beberapa terminal blok yang disediakan guna mengatur *Volume*, *Bass*, *Treble*, dan *Mute* yang dihubungkan langsung ke *Arduino Mega 2560* agar dapat dikontrol. Dalam rangkaian tersebut terdapat tambahan rangkaian *transistor optocoupler* sebagai pemisah tegangan yaitu tegangan dari *Arduino Mega 2560* dan tegangan dari *Power Supply*, dan sebelum masuk ke *transistor optocoupler* terdapat suatu resistor sebesar 330 Ω guna memproteksi led yang ada di dalam *transistor*. Terlihat spesifikasi rangkaian MAXIM5406 pada Tabel 4.5.

Tabel 4.5 Spesifikasi Rangkaian MAXIM5406 Channel 1

Supply Input	Kondisi Awal		Maksimum			V Mute
	In	Out	Volume	Bass	Treble	
5,06 V	2,4 V	2,8 V	2,4V	2,4V	2,4V	2,4V



Gambar 4.14 Input Output MAXIM5406 Channel 1

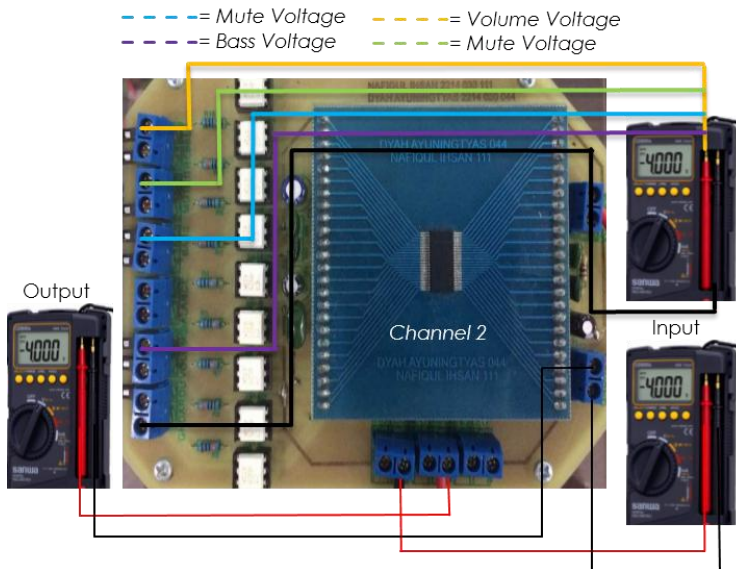
Dalam Gambar 4.14 pada monitor *oscilloscope* terlihat gelombang *input* yang masuk serta gelombang *output* yang di keluarkan oleh rangkaian MAXIM5406 dengan *input microphone dynamic* terlihat dalam grafik penguatan yang terjadi yang relatif kecil pada rangkaian MAXIM5406. Untuk melakukan perhitungan seberapa penguatan yang dilakukan oleh rangkaian MAXIM5406 *channel 1* menggunakan Persamaan 3.2 :

$$\begin{aligned} &= 20 \log (2,4\text{V}/2,8\text{V}) \\ &= 20 \log 0,85 \\ &= 20 (-0,07) \\ &= -1,4 \text{ dB} \end{aligned}$$

Dari peerhitungan diatas terlihat pelemahan yang dilakukan oleh rangkaian MAXIM5406 adalah sebesar -1,4 dB.

4.6 Pengujian Rangkaian MAXIM 5406 Channel 2

Pada rangkaian MAXIM5406 *channel 2* memiliki konfigurasi yang sama pada *channel 1* Gambar 4.15 tampilan rangkaian MAXIM5406 *channel 2*.

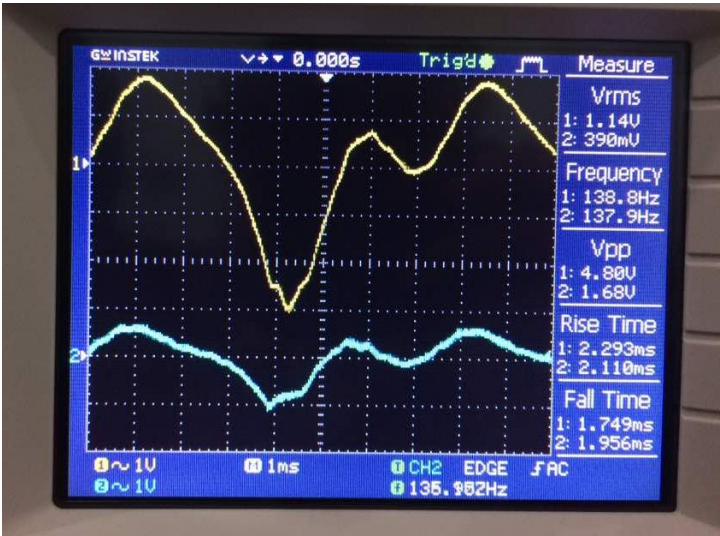


Gambar 4.15 Tone Control Digital MAXIM5406 Channel 2

Pada Gambar 4.15 terlihat rangkaian *tone control digital* MAXIM5406 yang konfigurasinya sama dengan rangkaian MAXIM5406 pada *channel 1*. Pada rangkaian MAXIM5406 terdapat beberapa tambahan sedikit kapasitor sebagai *filter* supaya dalam pengaturan *Volume*, *Treble*, *Bass*, dan *Mute* tidak terjadi *noise*. *Terminal* blok yang terdapat disisi kiri meliputi *Volume up*, *Volume down*, *Bass up*, *Bass down*, *Treble up*, *Treble down*, dan *Mute*. Dimana pada masing masing terminal blok masuk pada pin Arduino dengan nomor yang berbeda-beda untuk setiap konfigurasi, untuk melakukan *setting* audio.

Tabel 4.6 Spesifikasi Rangkaian MAXIM5406 *Channel 2*

Supply Input	Kondisi Awal		Maksimum			V Mute
	In	Out	Volume	Bass	Treble	
5,06 V	2,4 V	2,7 V	2,4V	2,4V	2,4V	2,4V



Gambar 4.16 Input Output MAXIM5406 *Channel 2*

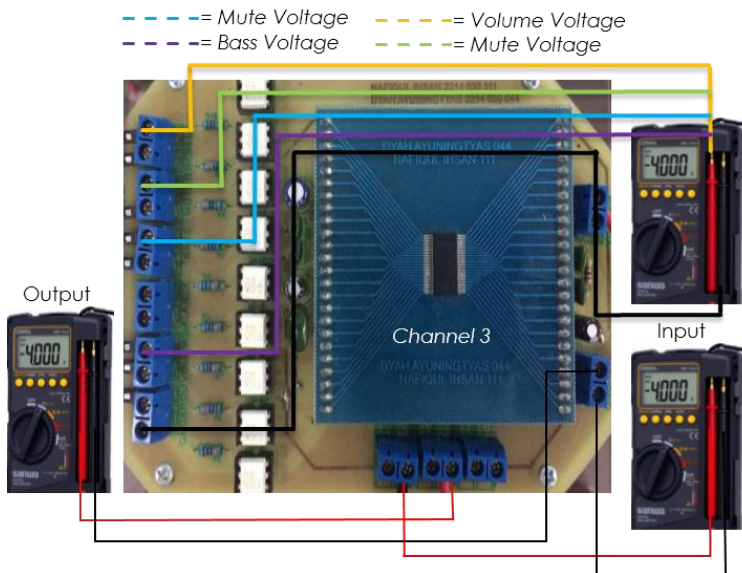
Tampilan monitor *oscilloscope* yang ditunjukkan oleh Gambar 4.16 terlihat bahwa terjadi pelemahan pada V_{rms} maupun V_{pp} nya dinama yang berwarna kuning merupakan gelombang *input* sedangkan warna biru merupakan gelombang *output* yang dihasilkan oleh rangkaian MAXIM5406. Untuk pengukuran pelemahannya maka menggunakan Persamaan 3.2:

$$\begin{aligned}
 &= 20 \log (2,4V/2,8V) \\
 &= 20 \log 0,85 \\
 &= 20 (-0,07) \\
 &= -1,4 \text{ dB}
 \end{aligned}$$

Dari perhitungan diatas terlihat pelemahan yang dilakukan oleh rangkaian MAXIM5406 adalah sebesar -1,4 dB.

4.7 Pengujian Rangkaian MAXIM 5406 Channel 3

Pada *input channel 3* rangkaian MAXIM 5406 pada Gambar 4.17 mendapat *input* dari keluaran *Pre-Amp* yang berasal dari gitar listrik. Untuk konfigurasi rangkaian MAXIM5406 *channel 3* ini mirip dengan *channel 1* dan *channel 2*.

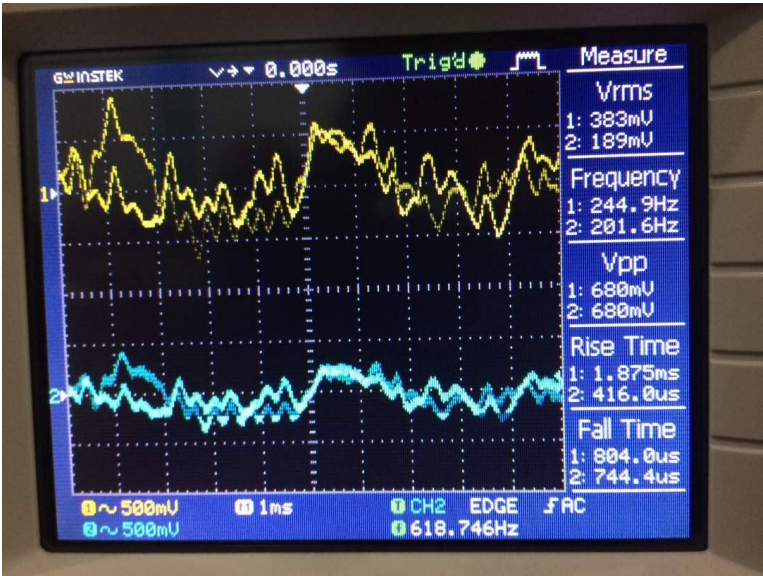


Gambar 4.17 Tone Control Digital MAXIM5406 Channel 3

Pada Gambar 4.17 terlihat rangkaian *tone control digital* dengan MAXIM5406.dimana konfigurasinya sama dengan *channel* sebelumnya yang sisi kiri terdapat terminal *block* guna mengatur selera audio menggunakan Arduino Mega 2560 kemudian sebelah kanan merupakan *input supply* dari *power supply* dimana yang kabel merah merupakan terminal positif dan kabel coklat merupakan terminal negatif. Dan sisi bawah terdapat 3 terminal blok yang berfungsi sebagai *input* rangkian, serta *output* dari rangkaian. Setiap *channel* terdapat *transistor optocoupler* sebagai peremisah tegangan agar dapat dilakukan pengaturan. Spesifikasi rangkian MAXIM5406 terlihat pada Tabel 4.7.

Tabel 4.7 Spesifikasi Rangkaian MAXIM5406 Channel 3

Supply Input	Kondisi Awal		Maksimum			V Mute
	In	Out	Volume	Bass	Treble	
5,06 V	2,4 V	2,8 V	2,4V	2,4V	2,4V	2,4V



Gambar 4.18 Input Output MAXIM5406 Channel 3

Pada Gambar 4.18 yang menampilkan layar *oscilloscope* terlihat sedikit pelemahan yang terjadi dari sinyal *input* dan *ouput*. Terlihat gelombang berwarna kuning yang menunjukkan *input* mengalami suatu pelemahan yang semula sebesar 383 mV kemudian ketika keluar menjadi 189 mV yang ditunjukkan oleh gelombang berwarna biru sebagai *output* dari sinyal audionya. Namun pada *Vpp* nya tidak terjadi perubahan yaitu sebesar 680 mV. Untuk menghitung pelemahan yang dilakukan oleh rangkaian MAXIM5406 maka menggunakan persamaan rumus :

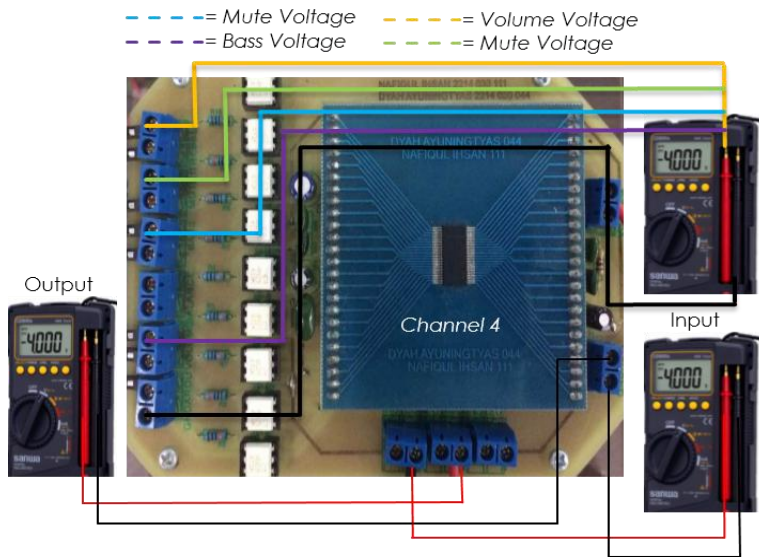
$$\begin{aligned} &= 20 \log (2,4\text{V}/2,8\text{V}) \\ &= 20 \log 0,85 \\ &= 20 (-0,07) \\ &= -1,4 \text{ dB} \end{aligned}$$

Dari perhitungan diatas terlihat pelemahan yang dilakukan oleh rangkaian MAXIM5406 adalah sebesar -1,4 dB. Persamaan tersebut berasal dari penurunan tegangan yang terjadi pada rangkaian MAXIM5406 di *channel 3*.

4.8 Pengujian Rangkaian MAXIM5406 Channel 4

Untuk konfigurasi rangkaian MAXIM5406 *Channel 4* ini memiliki konfigurasi yang sama dengan *channel-channel* sebelumnya yang terlihat pada Gambar 4.19, namun *channel 4* ini mendapat *input* dari *Pre-Amp* yang berasal dari *Keyboard Instrument*,serta mendapat *supply* *iinput* sebesar 5,06 V.

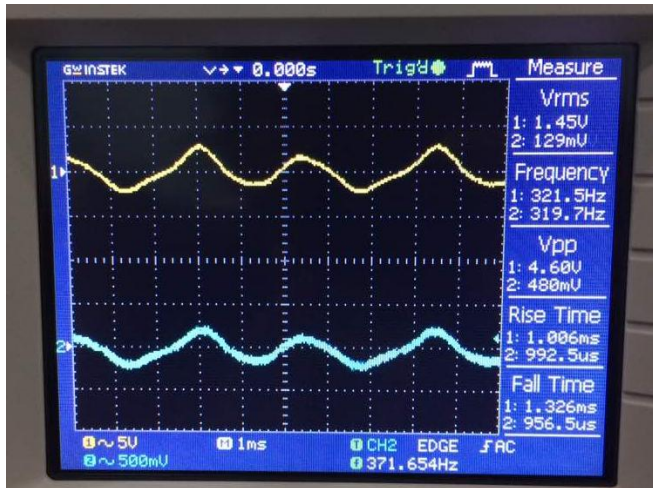
Pada Gambar 4.19 terlihat rangkaian MAXIM5406 *channel 4* yang mendapatkan *input* berupa *keyboard instrument*. konfigurasi dari *channel* ini mirip dengan konfigurasi dari *channel* sebelumnya yaitu terdapat beberapa terminal blok serta *transistor optocoupler* serta beberapa *transistor* dengan kapasistas kecil yang berperan sebagai *filter* audio agar tidak begitu terjadi *noise*. dalam proses menghubungkannya ke kontroller Arduino Mega 2560 menggunakan kabel pelangi yang dipadukan dengan kabel serabut biasa. Serta peran resistor diatas adalah melakukan proteksi terhadap led yang terdapat di dalam *transistor optocoupler* agar tidak terjadi *over voltage* yang menyebabkan led rusak.



Gambar 4.19 Tone Control Digital MAXIM5406 Channel 4

Tabel 4.8 Spesifikasi Rangkaian MAXIM 5406 Channel 4

Supply Input	Kondisi Awal		Maksimum			V Mute
	In	Out	Volume	Bass	Trebbles	
5,06 V	2,4 V	2,8 V	2,4V	2,4V	2,4V	2,4V



Gambar 4.20 Input Output MAXIM 5406 Channel 4

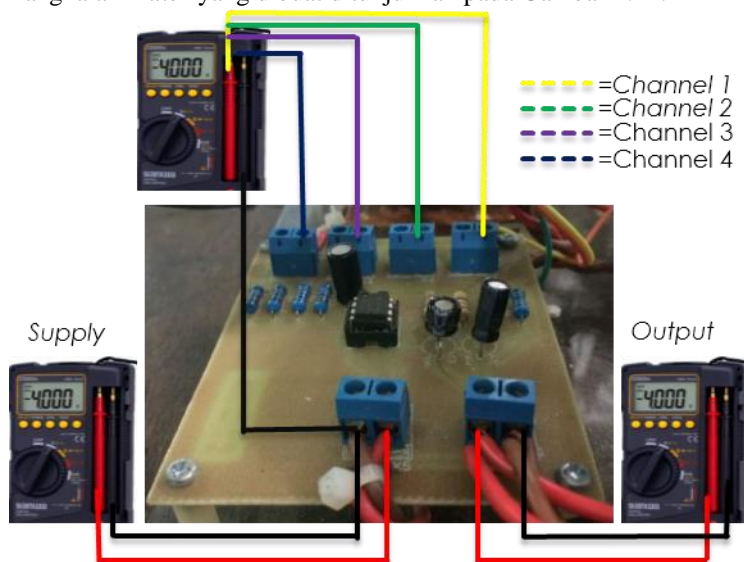
Pada Gambar 4.20 terlihat suatu pelemahan *input* yang dilakukan oleh rangkaian MAXIM5406 yaitu untuk tegangan efektifnya yang semula sebesar 1,45 V menjadi 129 mV terjadi suatu pelemahan, begitu juga dengan *voltage peek to peek*. Pelemahan ini terjadi dikarenakan *setup datasheet* dari MAXIM5406 yang hanya dapat menghasilkan sinyal audio dengan tegangan 2,4-2,5 V saja. Kemudian jika ingin menambahkan besar tegangan output maka kita dapat bermain pada rangkaian *Amplifier* yang berperan menerima keluaran dari sinyal audio MAXIM5406. Untuk perhitungan pelemahannya maka menggunakan Persamaan 3.2 :

$$\begin{aligned}
 &= 20 \log (2,4\text{V}/2,8\text{V}) \\
 &= 20 \log 0,85 \\
 &= 20 (-0,07) \\
 &= -1,4 \text{ dB}
 \end{aligned}$$

Dari perhitungan diatas terlihat pelemahan yang dilakukan oleh rangkaian MAXIM5406 adalah sebesar -1,4 dB. Pelemahan terjadi karena tegangann *input* lebih besar dari tegangan *output* yang terjadi pada rangkaian MAXIM5406 *channel 4*.

4.9 Pengujian Rangkaian Mixer

Pada rangkaian *mixer* ini menggunakan IC *oph-amp* dengan tipe TL081CN dengan perbandingan antara R1 dan R2 memiliki perbandingan yang sama yaitu 1:1. Serta fungsi rangkaian ini adalah menggabungkan masing-masing *output* dari IC MAXIM5406 menjadi satu serta dikeluarkan secara bersamaan pada *Amplifier*. Kami menggunakan *sound amplifier mini fender* sebagai *output* dari keseluruhan sistem. Dalam penggunaan IC TL081CN dalam rangkaian *mixer* ini menggunakan penguatan secara *inverting* dimana pin *negative* pada IC mendapat *feedback* dari pin *output* dari IC TL081CN ini. Untuk *output* dari rangkaian *mixer* ini yaitu terdapat pada 1 terminal blok yang saya paralel langsung agar dapat masuk kedalam 2 jenis *jack* dengan ukuran yang berbeda yaitu berukuran 5 mm *jack* dan 3 mm *jack*. Rangkaian *Mixer* yang dibuat ditunjukkan pada Gambar 4.21.



Gambar 4.21 Rangkaian Mixer

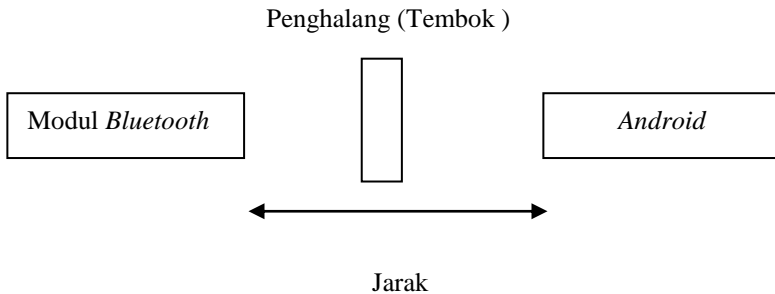
Tabel 4.9 Spesifikasi Rangkaian *Mixer*

Tegangan <i>Supply</i>	Tegangan <i>Input</i>				Tegangan <i>Output</i>
	CH1	CH2	CH3	CH4	
12,36 V	2,83 V	2,78 V	2,84 V	2,63 V	2,76 V

Dalam rangkaian *mixer* yang tampilan pada Gambar 4.21 merupakan rangkaian yang menggabungkan beberapa *output* audio dari rangkaian MAXIM5406 yang masuk sebagai *input* dan kemudian akan dikeluarkan menjadi satu pada satu terminal blok yang dihubungkan dengan 2 *jack input* 3 mm dan 5 mm dengan spesifikasi yang ditunjukkan pada Tabel 4.9.

4.10 Pengujian Status *Bluetooth* dan Pengiriman Data Terhadap Jarak Jangkau *Bluetooth*

Pengujian ini dilakukan bertujuan untuk mengetahui status *Bluetooth* dan pengiriman data terhadap jarak jangkau *Bluetooth*. Setelah dilakukan pengujian dengan halangan dan tanpa halangan, data hasil Status *Bluetooth* dan Pengiriman Data terhadap jarak jangkau *Bluetooth* yang diambil pada tanggal 7 Juni 2017 pukul 23.00. Pengujian status *Bluetooth* dan pengiriman data terhadap jarak jangkau *Bluetooth* dengan halangan ditunjukkan pada Gambar 4.22 dan hasil pegujian ditunjukkan pada Tabel 4.10. Pada Gambar 4.22 menunjukkan bahwa terdapat penghalang berupa tembok, tembok tersebut tetap terhitung dalam jarak. Jarak yang tercantum merupakan jarak terdekat yaitu 1 meter. Jarak yang menjadi jarak maksimal merupakan jarak ketika *Bluetooth* tidak terhubung seperti yang ditunjukkan pada Gambar 3.16 pada Bab III.



Gambar 4.22 Pengujian Jarak Jangkauan *Bluetooth* dengan Halangan

Tabel 4.10 Pengujian dengan Halangan

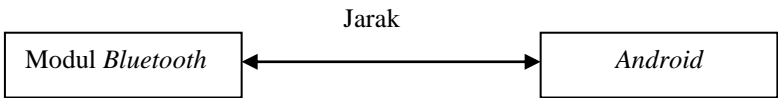
No.	Jarak	Status <i>Bluetooth</i>	Pengiriman Data
1	1 Meter	Terhubung	Terkirim
2	2 Meter	Terhubung	Terkirim
3	3 Meter	Terhubung	Terkirim
4	4 Meter	Terhubung	Terkirim
5	5 Meter	Terhubung	Terkirim
6	6 Meter	Terhubung	Terkirim
7	7 Meter	Terhubung	Terkirim
8	8 Meter	Terhubung	Terkirim
9	8,5 Meter	Terhubung	Terkirim
10	9 Meter	Tidak Terhubung	Tidak Terkirim
11	10 Meter	Tidak Terhubung	Tidak Terkirim

Pengujian status *bluetooth* dan pengiriman data terhadap jarak jangkauan *bluetooth* dengan penghalang didapat dari status yang tampil pada *interface* yang terdapat pada *Android* dengan indikator yang akan tampil seperti pada Gambar 4.24 pengiriman data yang ditunjukkan seperti Gambar 4.27.

Pada data yang disajikan pada Tabel 4.10 ditunjukkan bahwa pengiriman data tidak terkirim pada jarak 9 meter ketika diberikan halangan berupa tembok. Status *bluetooth* akan mempengaruhi

keberhasilan pengiriman data. Ketika *Bluetooth* terputus maka pengiriman data otomatis tidak terkirim.

Pada Gambar 4.23 ditunjukkan pengujian status *Bluetooth* dan pengiriman data terhadap jarak jangkauan *Bluetooth* tanpa halangan dan hasil pengujian ditunjukkan pada Tabel 4.11.



Gambar 4.23 Pengujian Jarak Jangkauan *Bluetooth* Tanpa Halangan

Tabel 4.11 Pengujian Tanpa Halangan

No.	Jarak	Status <i>Bluetooth</i>	Pengiriman Data
1	1 Meter	Terhubung	Terkirim
2	2 Meter	Terhubung	Terkirim
3	3 Meter	Terhubung	Terkirim
4	4 Meter	Terhubung	Terkirim
5	5 Meter	Terhubung	Terkirim
6	6 Meter	Terhubung	Terkirim
7	7 Meter	Terhubung	Terkirim
8	8 Meter	Terhubung	Terkirim
9	9 Meter	Terhubung	Terkirim
10	10 Meter	Terhubung	Terkirim
11	11 Meter	Terhubung	Terkirim
12	12 Meter	Terhubung	Terkirim
13	13 Meter	Terhubung	Terkirim
14	14 Meter	Terhubung	Terkirim
15	15 Meter	Terhubung	Terkirim
16	16 Meter	Terhubung	Terkirim
17	17 Meter	Terhubung	Terkirim
18	18 Meter	Terhubung	Terkirim
19	19 Meter	Terhubung	Terkirim
20	20 Meter	Terhubung	Terkirim
21	21 Meter	Terhubung	Terkirim
22	22 Meter	Terhubung	Terkirim
23	23 Meter	Terhubung	Terkirim
24	25 Meter	Terhubung	Terkirim
25	25,5 Meter	Tidak Terhubung	Tidak Terkirim
26	26 Meter	Tidak Terhubung	Tidak Terkirim
27	27 Meter	Tidak Terhubung	Tidak Terkirim

Pengujian status *bluetooth* dan pengiriman data terhadap jarak jangkauan *bluetooth* tanpa penghalang didapat dari status yang tampil pada *interface* yang terdapat pada *Android* dengan indikator yang akan tampil seperti pada Gambar 4.24 dan pengiriman data yang ditunjukkan seperti Gambar 4.27.

Pada data yang disajikan pada Tabel 4.11 ditunjukkan bahwa pengiriman data tidak terkirim pada jarak 25,5 meter tanpa halangan. Begitu juga pada data ini, status *Bluetooth* akan mempengaruhi keberhasilan pengiriman data, ketika *Bluetooth* terputus secara otomatis data tidak akan terkirim.

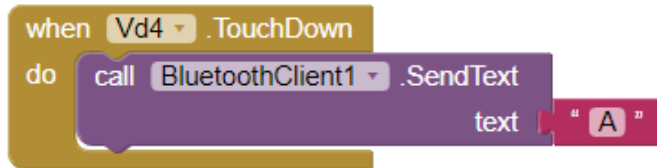


Gambar 4.24 Tampilan Aplikasi Ketika Terhubung

4.11 Pengujian Pengiriman Data *Android* ke Arduino Mega 2560

Pengujian ini bertujuan untuk mengetahui apakah aplikasi *Android* yang dibuat dapat mengirimkan data teks ke Arduino Mega 2560 dengan sesuai atau tidak. Pengujian diamati melalui serial monitor yang ada pada *Software* Arduino. Hasil pengiriman data akan ditunjukkan pada Gambar 4.25 dan Gambar 4.26. Pengambilan data diwakili dengan salah

satu pengiriman data yaitu tombol untuk mengurangi *Volume* pada *channel 4* yang ditandai dengan pengiriman data huruf “A”.



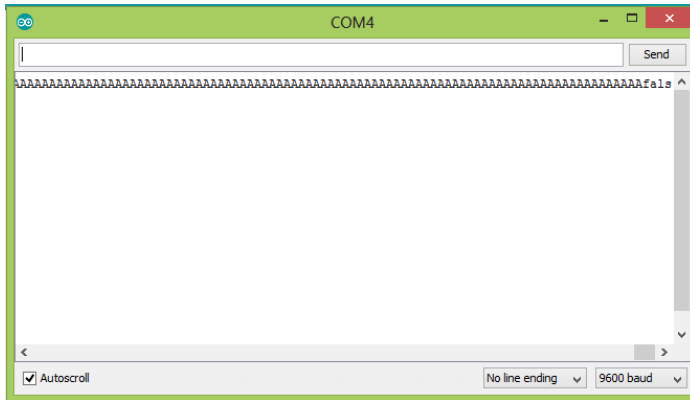
Gambar 4.25 Data yang Akan Dikirim oleh Aplikasi *Android*

Dengan pengiriman data tersebut, agar lebih mudah untuk mengetahui data yang dikirim maka pada program perintah pengiriman data dan hasil pengiriman disamakan yaitu sama-sama dengan menggunakan huruf “A” seperti yang ditunjukkan pada Gambar 4.24.

```
if (val == 'A'){  
    digitalWrite(23,HIGH);  
    delayMicroseconds(1000);  
    digitalWrite(23,LOW);  
    Serial.print("A");  
}
```

Gambar 4.26 Program Untuk Mengirim Data “A”

Kemudian untuk memastikan data yang dikirimkan oleh *Android* diterima oleh *Arduino Mega 2560*, maka dapat dipantau melalui *Serial Monitor* yang tersedia pada *Arduino Software*. Hasil pantauan melalui *Serial Monitor* dapat dilihat pada Gambar 4.27 yang kemudian hasil dari pantauan masing-masing teks setiap *tone control* pada *Serial Monitor* akan terbaca seperti pada Tabel 4.12.



Gambar 4.27 Data yang Dibaca oleh Arduino Mega 2560

Tabel 4.12 Data Teks yang Dibaca oleh Arduino

No.	Teks Data yang Dikirim	Keterangan
1	a	Terbaca
2	b	Terbaca
3	c	Terbaca
4	d	Terbaca
5	G	Terbaca
6	f	Terbaca
7	g	Terbaca
8	h	Terbaca
9	i	Terbaca
10	j	Terbaca
11	k	Terbaca
12	l	Terbaca
13	m	Terbaca
14	n	Terbaca
15	o	Terbaca
16	p	Terbaca
17	q	Terbaca
18	r	Terbaca
19	s	Terbaca
20	t	Terbaca
21	u	Terbaca
22	v	Terbaca
23	w	Terbaca
24	x	Terbaca

No.	Teks Data yang Dikirim	Keterangan
25	y	Terbaca
26	z	Terbaca
27	A	Terbaca
28	B	Terbaca

Pada Tabel 4.12 ditunjukkan bahwa data yang dikirim oleh *Android* sesuai dan terbaca seluruhnya oleh Arduino yang dipantau melalui *Serial Monitor Arduino Software*.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Setelah melakukan perencanaan, perancangan, dan pengujian alat maka dapat diambil kesimpulan dan memberikan saran demi penyempurnaan Tugas Akhir ini.

5.1 Kesimpulan

Hasil dari pengujian serta analisa data dari Status *Bluetooth* dan Pengiriman Data terhadap Jarak Jangkauan *Bluetooth* dapat disimpulkan bahwa:

1. Rangkaian *Mixer* memiliki penguatan yang perbandingannya sama yaitu 1:1.
2. Kondisi *On Volume* rangkaian MAXIM5406 sebesar -20dB dan maksimal -59dB.
3. Pengiriman data ketika diberi halangan pada jarak 9 meter tidak terkirim.
4. Pengiriman data ketika tanpa halangan pada jarak 25,5 meter tidak terkirim.
5. Untuk menambah jumlah *channel* maka dibutuhkan tambahan MAXIM5406.

5.2 Saran

Untuk pengembangan alat selanjutnya sebaiknya alat tersebut di beri *interface* atau monitor digital untuk melihat status dari IC *processor* dalam kondisi *on* atau *off* serta mengetahui sedang pada tingkat berapa kondisi untuk *Volume*, *Treble*, dan *Bass*. Untuk komunikasinya menggunakan modul yang lebih mumpuni terhadap jarak, agar dapat dikontrol pada jarak yang lebih jauh. Serta untuk jumlah *channel* dari *mixer* agar ditambah lagi atau diperbanyak, sehingga dapat menampung beberapa alat *instrument music* lainnya.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1]“ *Perekayasaan Sistem Audio*”. Kementrian Pendidikan dan Kebudayaan:Jakarta. 2013
- [2]“*Datasheet Pre-Amp TL081CN*”:Atmel Corporation.San Jose USA. 2014
- [3]“*Datasheet SMD IC MAXIM 5406*’ Corporation.San Jose USA. 2014
- [4] Kuhn, Markus.”*Digital Signal Prosessing*”. Cambridgeshire:University Of Cambridge. 2009
- [5],”*Arduino Mega 2560*”.
<https://www.Arduino.cc/en/Main/ArduinoBoardMega2560>, diakses tanggal 19 Juni 2017
- [6] Vishay.”*Optocoupler, Phototransistor Output, with Base Connection*” 2010
- [7] Putra,Rahmat Prayogi.”*Teknik Audio Video*”.Jakarta. 2008
- [8] Donald, Frank.” *Interfacing Bluetooth Module with 8051 Microcontroller (HC05)*”.Gadgetronicx:Tangerang. 2015
- [9] Sinduaidi. “*Mengenal Arduino Software IDE*’. Daerah Istimewa Yogyakarta. 2016
- [10],”*Designer and Blocks Editor*”.
”<http://AppInventor.MIT.edu/explore/designer-blocks.html>”.
Massachusetts Institute of Technology:Boston, diakses tanggal 10 Juni 2017
- [11] Delon.”*Oph-Amp Microphone Preamp*” Elektropage:Tangerang. 2007
- [12] Andrianto, Heri.”*Arduino Belajar Cepat* ”.Informatika:Bandung. 2016
- [13] Darmawan, Aan.”*Arduino Belajar Cepat dan Pemrograman*”. Penerbit Informatika.Bandung. 2016
- [14] Francis Rumsey dan Tim McCormick.”*Sound and Recording: An Introduction*”.Focal Press. Chapter 5:Mixers.2002
- [15],”*Serial*”.<https://www.Arduino.cc/en/serial/begin&prev=search>
Diakses tanggal 14 Juli 2017

-----Halaman ini sengaja dikosongkan-----

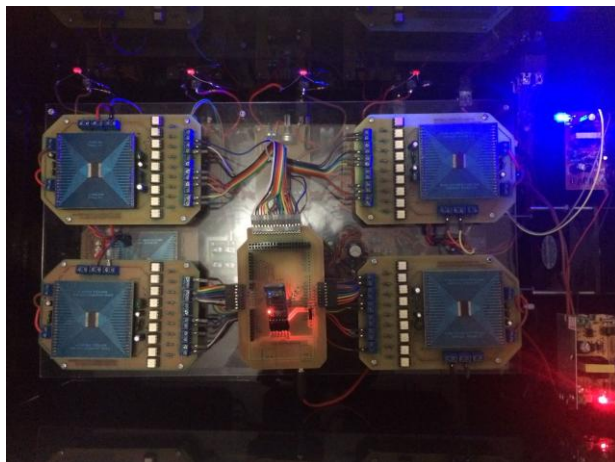
LAMPIRAN A

A.1. Tampilan Keseluruhan Rangkaian

- Tampak Depan *Box* Penutup Rangkaian Audio



- Tampilan Seluruh Rangkaian Tampak Atas



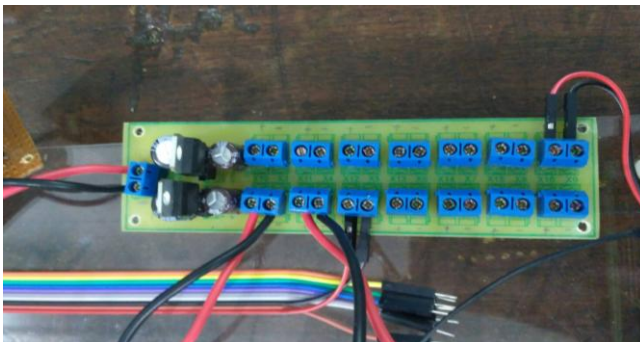
- *Power Supply 5Vdc 2A Sebagai Catu Arduino Mega 2560*



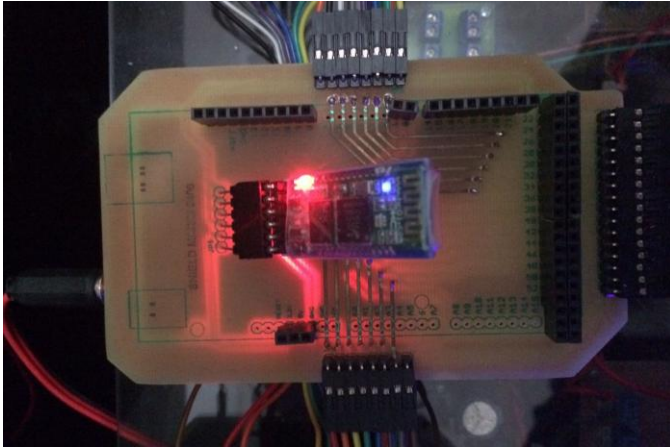
- *Power Supply 12Vdc 1A Sebagai Catu Fan*



- *Terminal 14 Pin Power Supply 12 V dan 5 V Sebagai Catu Rangkaian Pre-Amp, MAXIM5406 dan Mixer*



- Rangkaian Arduino Mega2560 With Shield dan Modul Bluetooth



- Fan Pendingin Seluruh Rangkaian Pada Box Mixer Audio



- Percobaan pada *Microphone Channel 1*



- Percobaan *Microphone Channel 2*



- Percobaan *Electric Gitar* Pada *Channel 3*



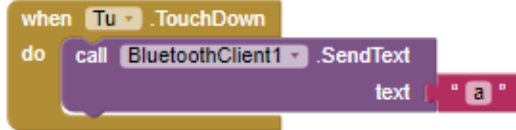
- Percobaan *Keyboard Instrument* pada *Channel 4*



A.2 Pengujian Pengiriman Data *Android* ke *Arduino Mega 2560*

1. Menambah *Treble* pada *Channel 1* dengan Data Pengiriman “a”

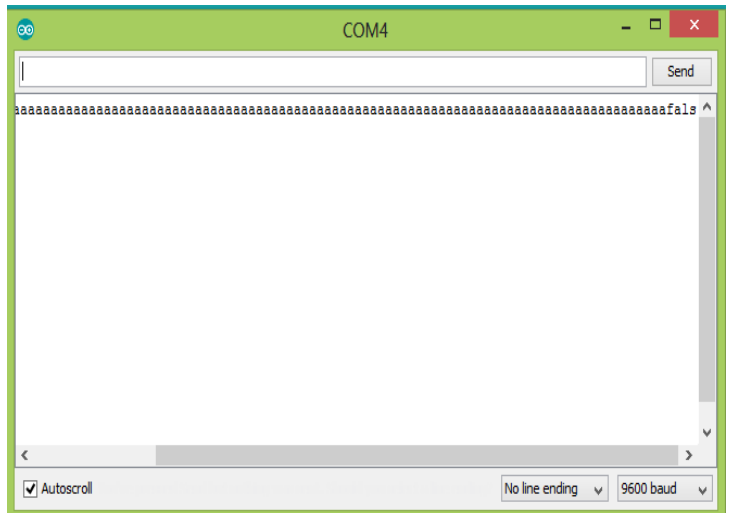
- Program *MIT App Inventor 2*



- Program *Arduino Software*

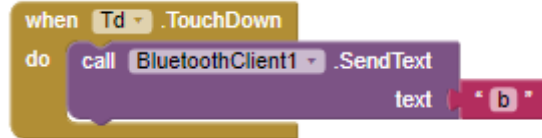
```
if (val == 'a') {  
    digitalWrite(37,HIGH);  
    delayMicroseconds(1000);  
    digitalWrite(37,LOW);  
    Serial.print("a");  
}
```

- *Serial Monitor*



2. Mengurangi *Treble* pada *Channel 1* dengan Data Pengiriman “b”

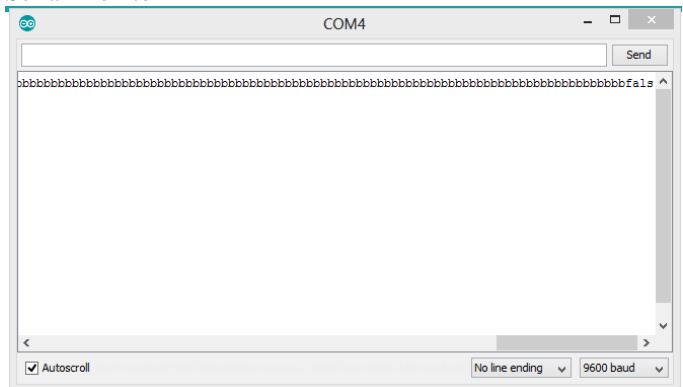
- Program *MIT App Inventor 2*



- Program *Arduino Software*

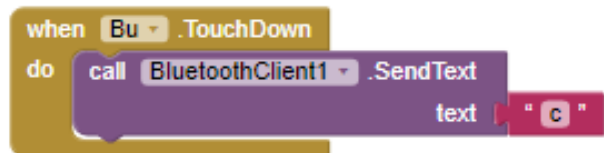
```
if (val == 'b'){  
    digitalWrite(39,HIGH);  
    delayMicroseconds(1000);  
    digitalWrite(39,LOW);  
    Serial.print("b");  
}
```

- *Serial Monitor*



3. Menambah *Bass* pada *Channel 1* dengan Data Pengiriman “c”

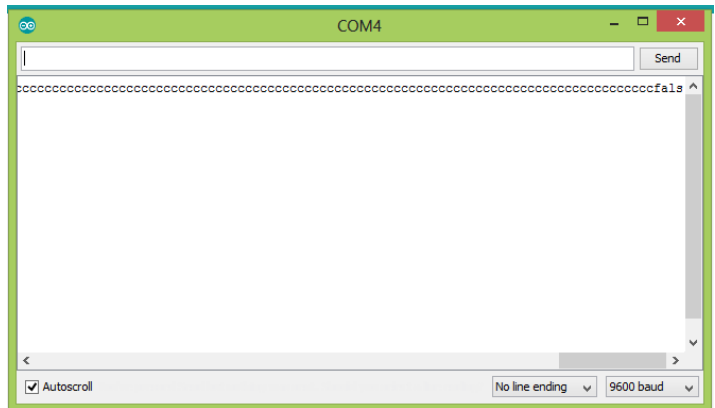
- Program *MIT App Inventor 2*



- Program Arduino *Software*

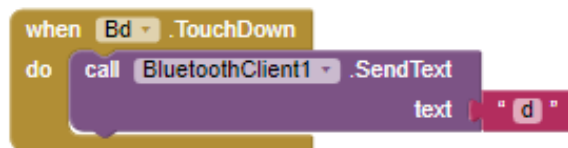
```
if (val == 'c'){
    digitalWrite(43,HIGH);
    delayMicroseconds(1000);
    digitalWrite(43,LOW);
    Serial.print("c");
}
```

- Serial Monitor



4. Mengurangi *Bass* pada *Channel 1* dengan Data Pengiriman “d”

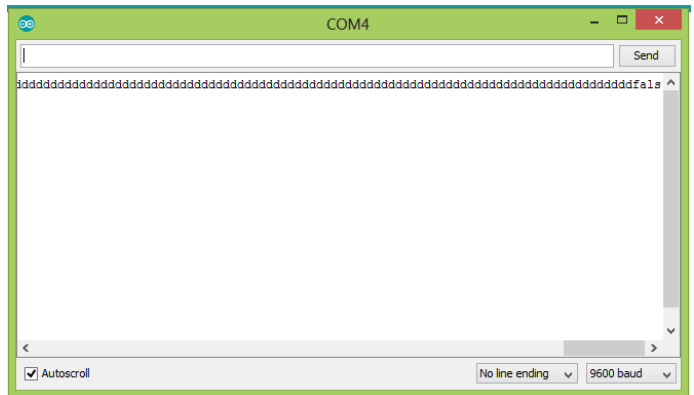
- Program MIT App Inventor 2



- Program Arduino *Software*

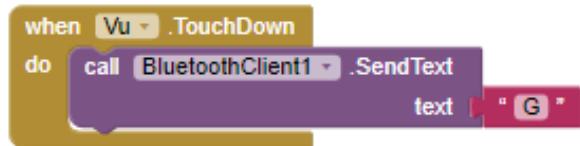
```
if (val == 'd'){
    digitalWrite(45,HIGH);
    delayMicroseconds(1000);
    digitalWrite(45,LOW);
    Serial.print("d");
}
```

- Serial Monitor



5. Menambah *Volume* pada *Channel 1* dengan Data Pengiriman "G"

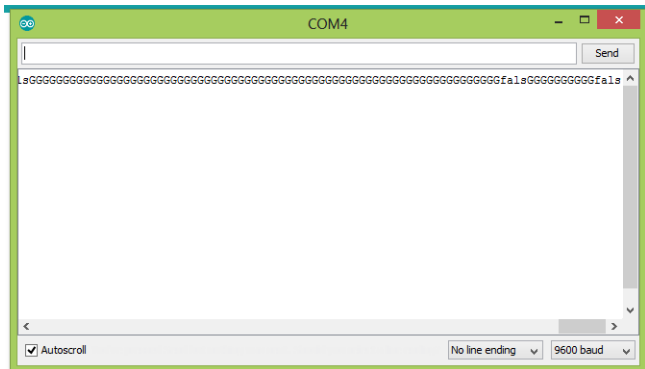
- Program *MIT App Inventor 2*



- Program *Arduino Software*

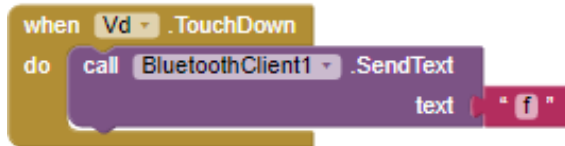
```
if (val == 'G'){
    digitalWrite(47,HIGH);
    delayMicroseconds(1000);
    digitalWrite(47,LOW);
    Serial.print("G");
}
```

- Serial Monitor



6. Mengurangi *Volume* pada *Channel 1* dengan Data Pengiriman "f"

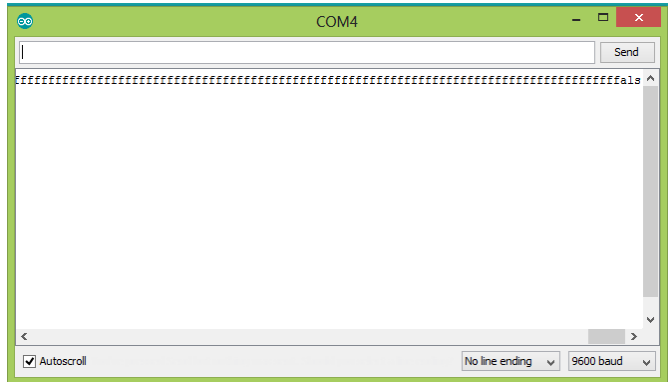
- Program *MIT App Inventor 2*



- Program *Arduino Software*

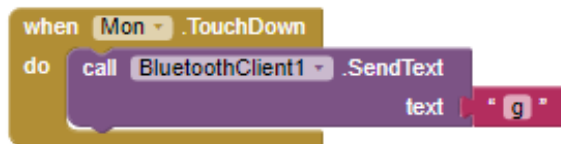
```
if (val == 'f') {
  digitalWrite(49, HIGH);
  delayMicroseconds(1000);
  digitalWrite(49, LOW);
  Serial.print("f");
}
```

- Serial Monitor



7. Melakukan *Mute* pada *Channel 1* dengan Data Pengiriman “g”

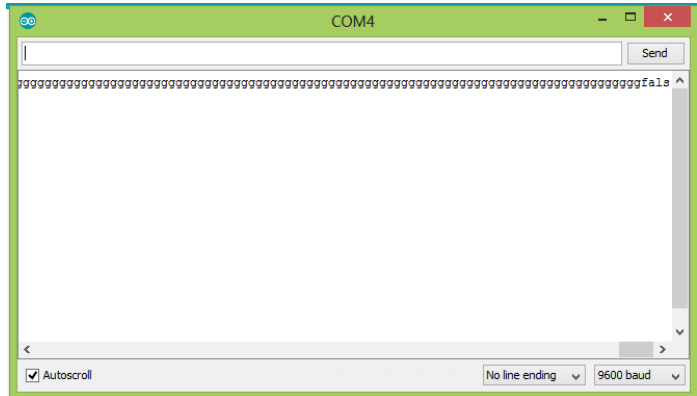
- Program *MIT App Inventor 2*



- Program *Arduino Software*

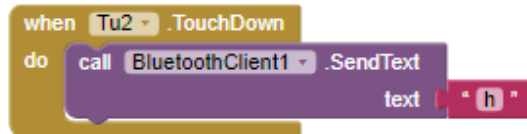
```
if (val == 'g'){
    digitalWrite(41,HIGH);
    delayMicroseconds(1000);
    digitalWrite(41,LOW);
    Serial.print("g");
}
```

- Serial Monitor



8. Menambah *Trebel* pada *Channel 2* dengan Data Pengiriman "h"

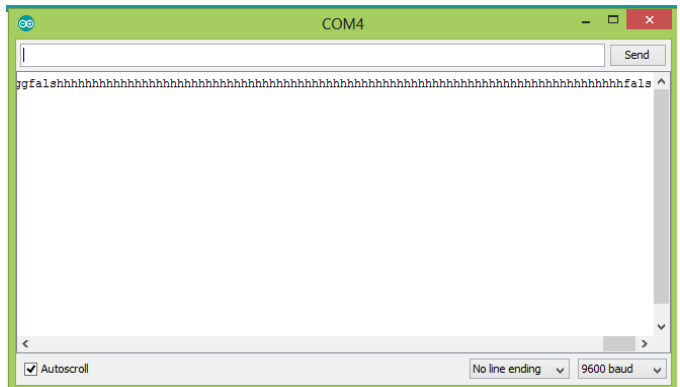
- Program *MIT App Inventor 2*



- Program *Arduino Software*

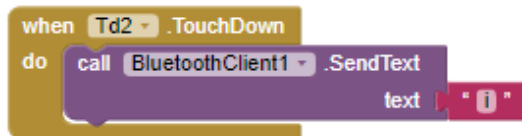
```
if (val == 'h'){
  digitalWrite(48,HIGH);
  delayMicroseconds(1000);
  digitalWrite(48,LOW);
  Serial.print("h");
}
```


- Serial Monitor



9. Mengurangi *Trebel* pada *Channel 2* dengan Data Pengiriman "i"

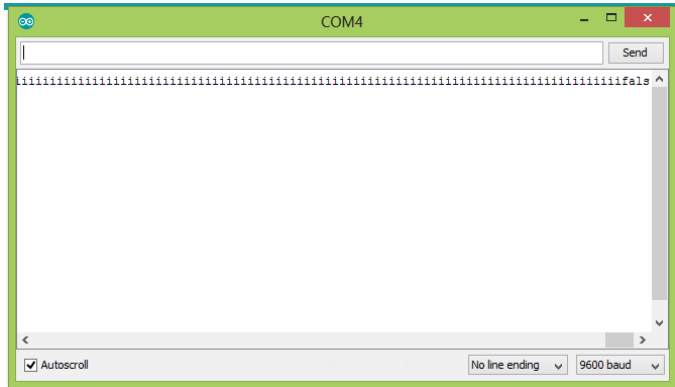
- Program *MIT App Inventor 2*



- Program *Arduino Software*

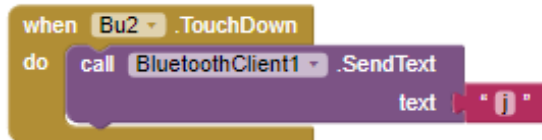
```
if (val == 'i'){
  digitalWrite(46,HIGH);
  delayMicroseconds(1000);
  digitalWrite(46,LOW);
  Serial.print("i");
}
```

- Serial Monitor



10. Menambah *Bass* pada *Channel 2* dengan Data Pengiriman “j”

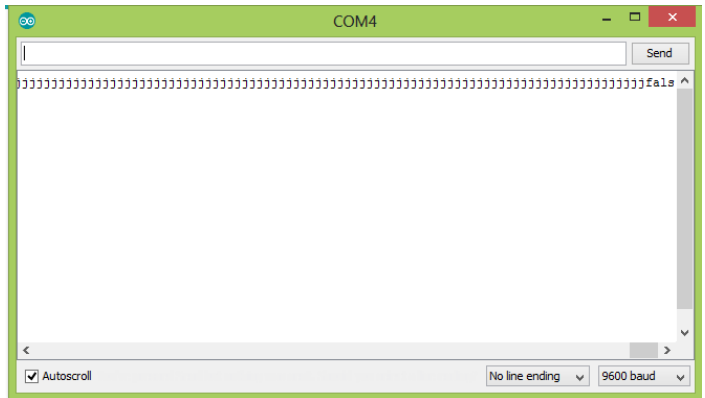
- Program *MIT App Inventor 2*



- Program *Arduino Software*

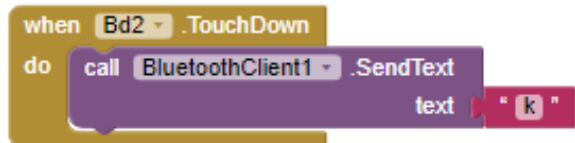
```
if (val == 'j'){
    digitalWrite(42,HIGH);
    delayMicroseconds(1000);
    digitalWrite(42,LOW);
    Serial.print("j");
}
```

- Serial Monitor



11. Mengurangi *Bass* pada *Channel 2* dengan Data Pengiriman “k”

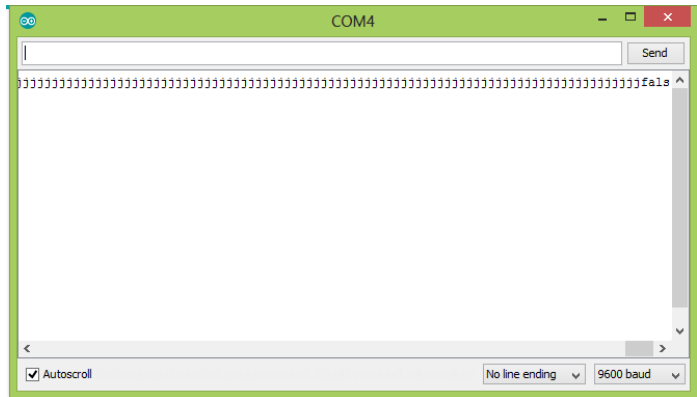
- Program *MIT App Inventor 2*



- Program *Arduino Software*

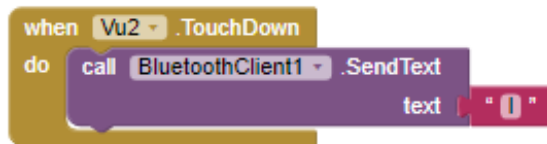
```
if (val == 'k'){
    digitalWrite(40,HIGH);
    delayMicroseconds(1000);
    digitalWrite(40,LOW);
    Serial.print("k");
}
```

- Serial Monitor



12. Menambah *Volume* pada *Channel 2* dengan Data Pengiriman "1"

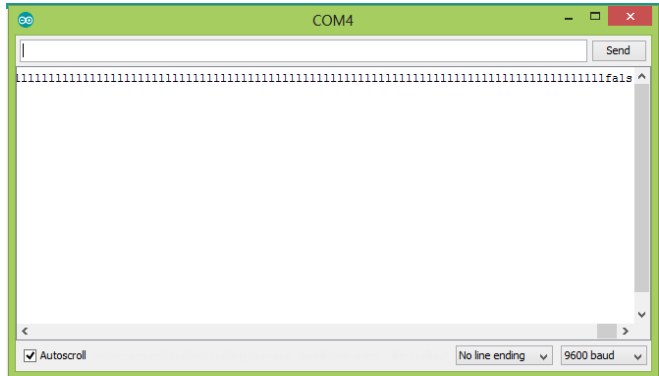
- Program *MIT App Inventor 2*



- Program *Arduino Software*

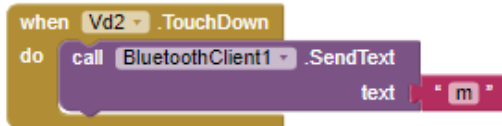
```
if (val == '1'){
  digitalWrite(38,HIGH);
  delayMicroseconds(1000);
  digitalWrite(38,LOW);
  Serial.print("1");
}
```

- Serial Monitor



13. Mengurangi *Volume* pada *Channel 2* dengan Data Pengiriman “m”

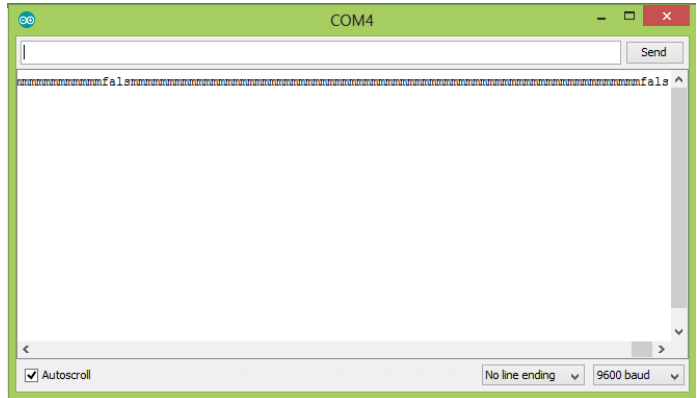
- Program MIT App Inventor 2



- Program Arduino Software

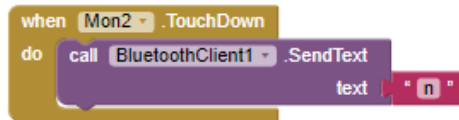
```
if (val == 'm'){
    digitalWrite(36,HIGH);
    delayMicroseconds(1000);
    digitalWrite(36,LOW);
    Serial.print("m");
}
```

- Serial Monitor



14. Melakukan *Mute* pada *Channel 2* dengan Data Pengiriman “n”

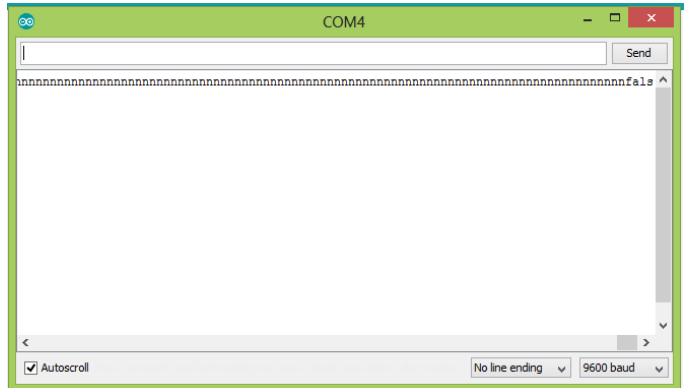
- Program *MIT App Inventor 2*



- Program *Arduino Software*

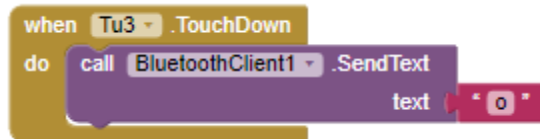
```
if (val == 'n') {
    digitalWrite(44, HIGH);
    delayMicroseconds(1000);
    digitalWrite(44, LOW);
    Serial.print("n");
}
```

- Serial Monitor



15. Menambah *Treble* pada *Channel 3* dengan Data Pengiriman “o”

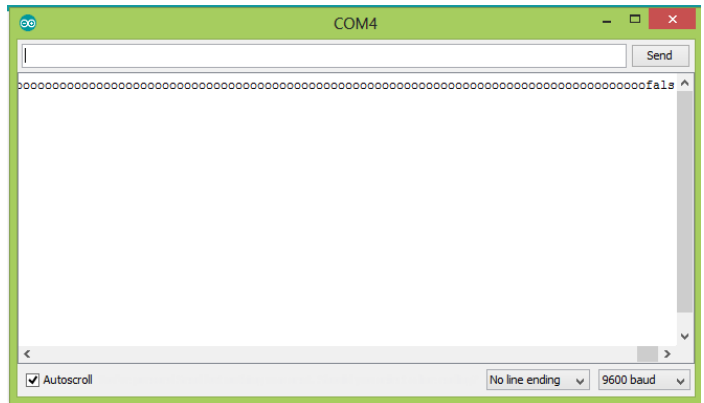
- Program *MIT App Inventor 2*



- Program *Arduino Software*

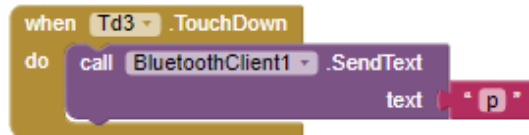
```
if (val == 'o'){
  digitalWrite(34,HIGH);
  delayMicroseconds(1000);
  digitalWrite(34,LOW);
  Serial.print("o");
}
```

- Serial Monitor



16. Mengurangi *Treble* pada *Channel 3* dengan Data Pengiriman “p”

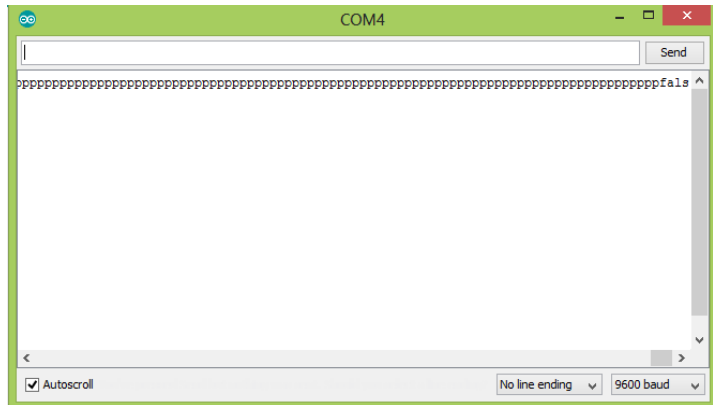
- Program *MIT App Inventor 2*



- Program *Arduino Software*

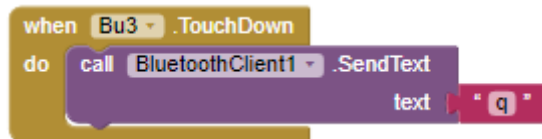
```
if (val == 'p'){
  digitalWrite(32,HIGH);
  delayMicroseconds(1000);
  digitalWrite(32,LOW);
  Serial.print("p");
}
```


- Serial Monitor



17. Menambah *Bass* pada *Channel 3* dengan Data Pengiriman “q”

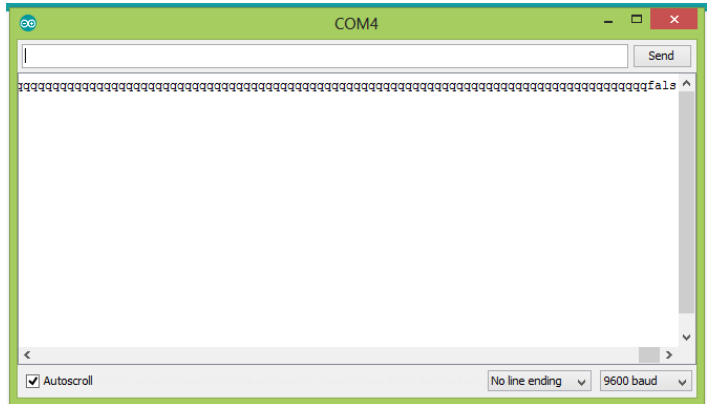
- Program *MIT App Inventor 2*



- Program *Arduino Software*

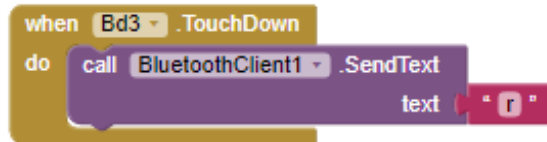
```
if (val == 'q'){
  digitalWrite(28,HIGH);
  delayMicroseconds(1000);
  digitalWrite(28,LOW);
  Serial.print("q");
}
```

- Serial Monitor



18. Mengurangi *Bass* pada *Channel 3* dengan Data Pengiriman “r”

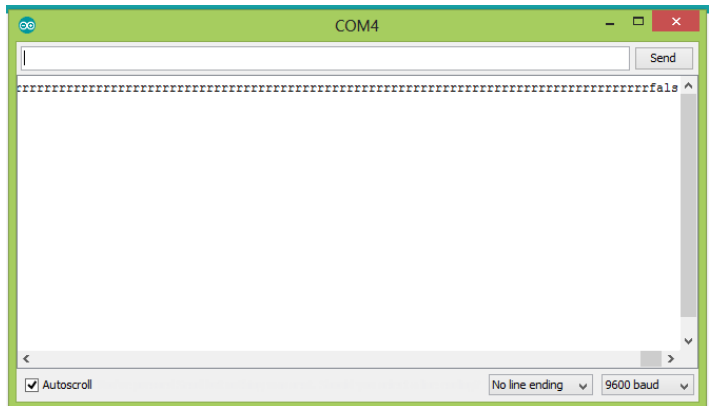
- Program *MIT App Inventor 2*



- Program *Arduino Software*

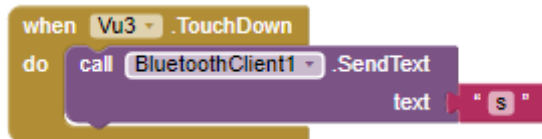
```
if (val == 'r'){
  digitalWrite(26,HIGH);
  delayMicroseconds(1000);
  digitalWrite(26,LOW);
  Serial.print("r");
}
```

- Serial Monitor



19. Menambah *Volume* pada *Channel 3* dengan Data Pengiriman "S"

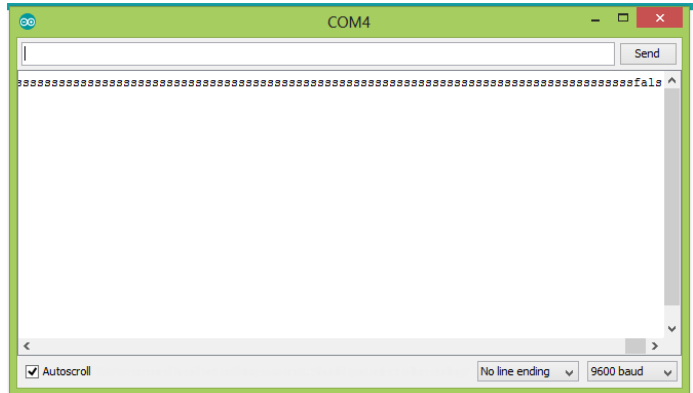
- Program *MIT App Inventor 2*



- Program *Arduino Software*

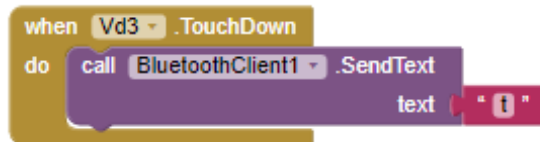
```
if (val == 's') {
  digitalWrite(24,HIGH);
  delayMicroseconds(1000);
  digitalWrite(24,LOW);
  Serial.print("s");
}
```

- Serial Monitor



20. Mengurangi *Volume* pada *Channel 3* dengan Data Pengiriman
“t”

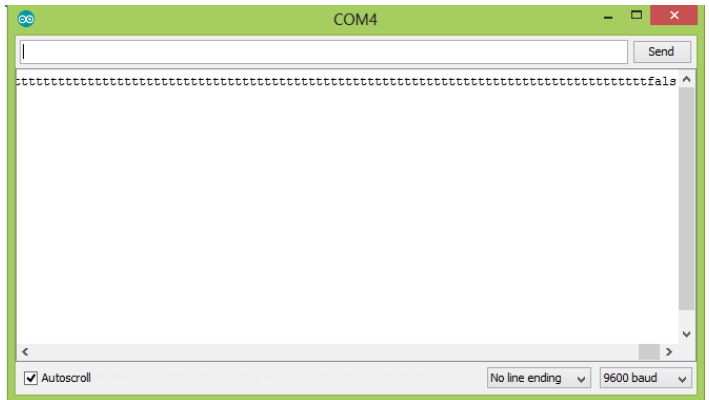
- Program *MIT App Inventor 2*



- Program *Arduino Software*

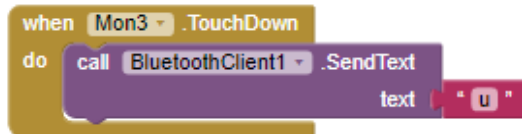
```
if (val == 't') {
  digitalWrite(22, HIGH);
  delayMicroseconds(1000);
  digitalWrite(22, LOW);
  Serial.print("t");
}
```

- Serial Monitor



21. Melakukan *Mute* pada *Channel 3* dengan Data Pengiriman “u”

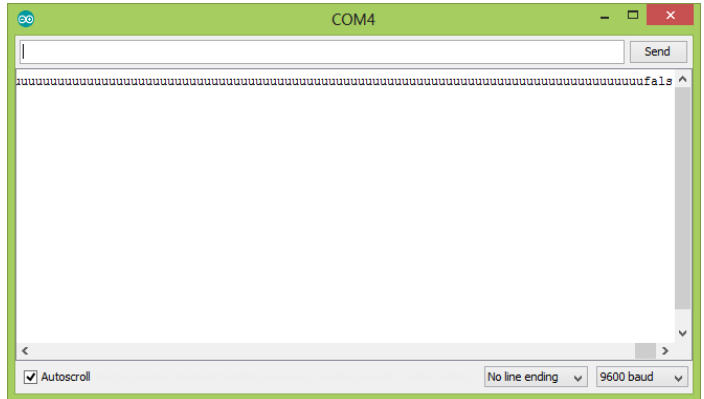
- Program *MIT App Inventor 2*



- Program *Arduino Software*

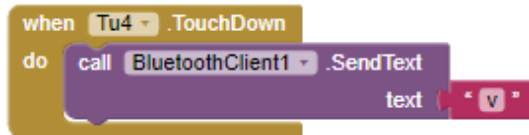
```
if (val == 'u') {
  digitalWrite(30,HIGH);
  delayMicroseconds(1000);
  digitalWrite(30,LOW);
  Serial.print("u");
}
```

- Serial Monitor



22. Menambah *Treble* pada *Channel 4* dengan Data Pengiriman
“v”

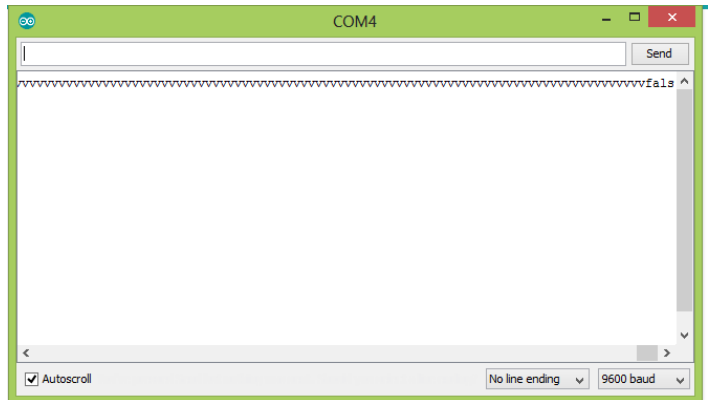
- Program MIT App Inventor 2



- Program Arduino Software

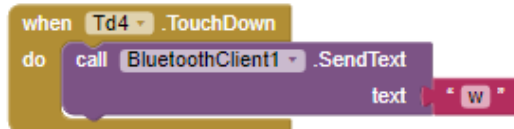
```
if (val == 'v') {
  digitalWrite(35, HIGH);
  delayMicroseconds(1000);
  digitalWrite(35, LOW);
  Serial.print("v");
}
```

- Serial Monitor



23. Mengurangi *Treble* pada *Channel 4* dengan Data Pengiriman “w”

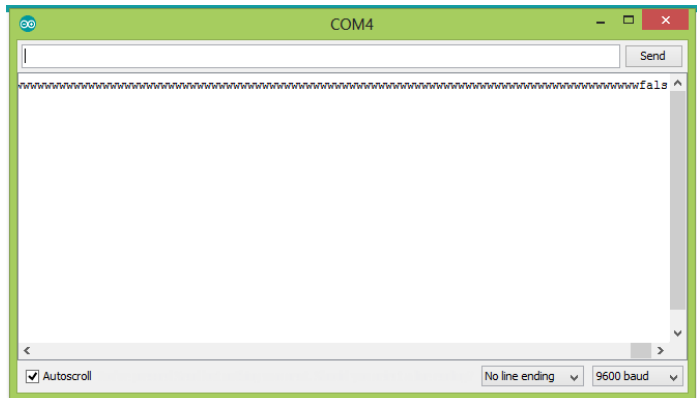
- Program *MIT App Inventor 2*



- Program *Arduino Software*

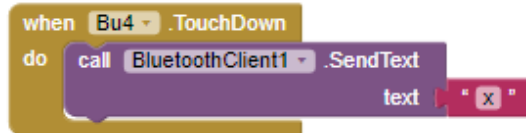
```
if (val == 'w'){
    digitalWrite(33,HIGH);
    delayMicroseconds(1000);
    digitalWrite(33,LOW);
    Serial.print("w");
}
```

- Serial Monitor



24. Menambah *Bass* pada *Channel 4* dengan Data Pengiriman “x”

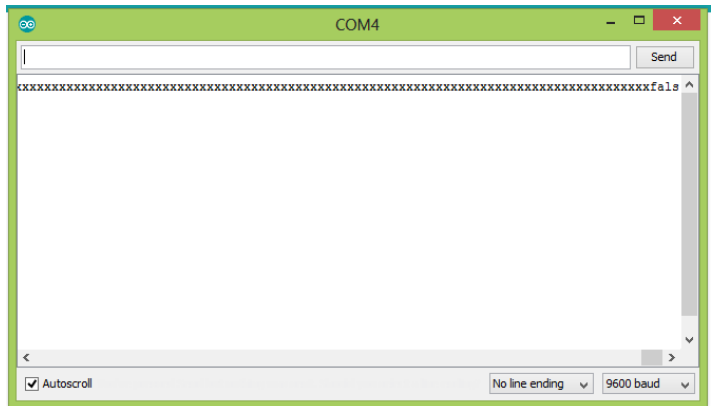
- Program *MIT App Inventor 2*



- Program *Arduino Software*

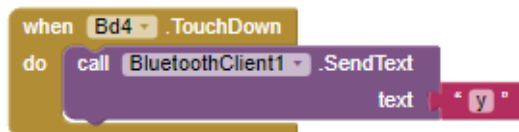
```
if (val == 'x') {
  digitalWrite(29,HIGH);
  delayMicroseconds(1000);
  digitalWrite(29,LOW);
  Serial.print("x");
}
```


- Serial Monitor



25. Mengurangi *Bass* pada *Channel 4* dengan Data Pengiriman “y”

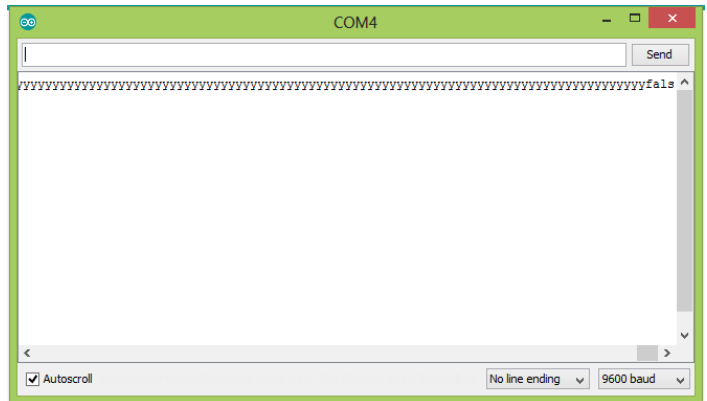
- Program *MIT App Inventor 2*



- Program *Arduino Software*

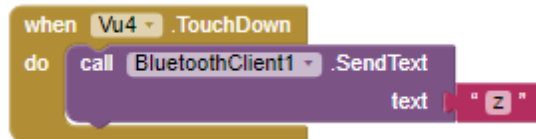
```
if (val == 'y'){
    digitalWrite(27,HIGH);
    delayMicroseconds(1000);
    digitalWrite(27,LOW);
    Serial.print("y");
}
```

- Serial Monitor



26. Menambah *Volume* pada *Channel 4* dengan Data Pengiriman
“Z”

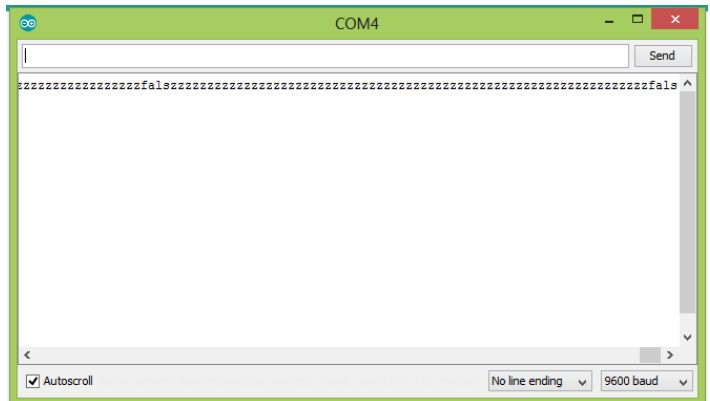
- Program *MIT App Inventor 2*



- Program *Arduino Software*

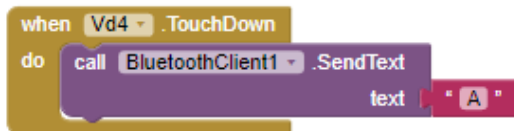
```
if (val == 'z'){
  digitalWrite(25,HIGH);
  delayMicroseconds(1000);
  digitalWrite(25,LOW);
  Serial.print("z");
}
```

- Serial Monitor



27. Mengurangi *Volume* pada *Channel 4* dengan Data Pengiriman "A"

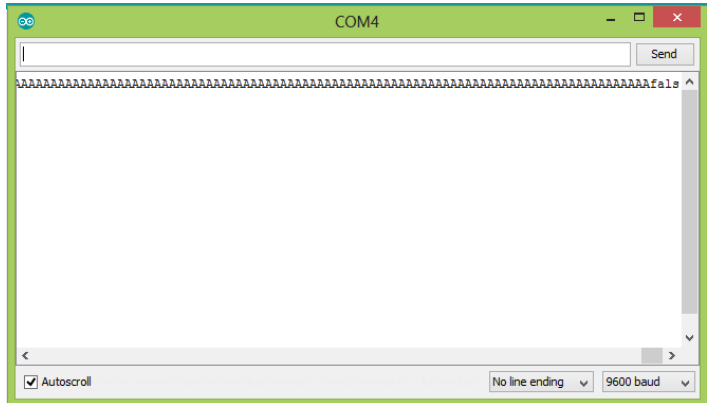
- Program *MIT App Inventor 2*



- Program *Arduino Software*

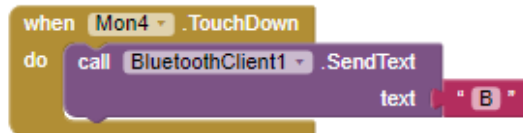
```
if (val == 'A'){
  digitalWrite(23,HIGH);
  delayMicroseconds(1000);
  digitalWrite(23,LOW);
  Serial.print("A");
}
```

- Serial Monitor



28. Melakukan *Mute* pada *Channel 4* dengan Data Pengiriman “A”

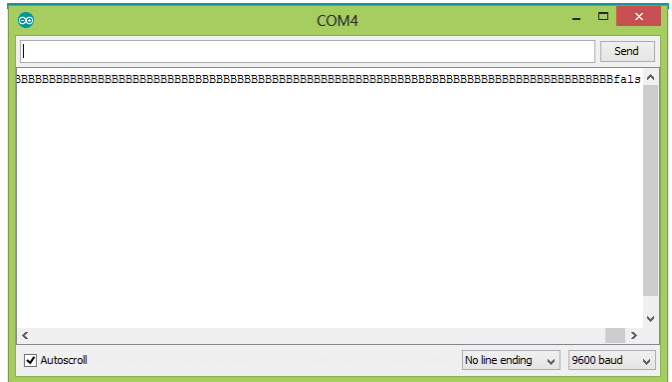
- Program *MIT App Inventor 2*



- Program *Arduino Software*

```
if (val == 'B'){
    digitalWrite(31,HIGH);
    delayMicroseconds(1000);
    digitalWrite(31,LOW);
    Serial.print("B");
}
```

- Serial Monitor



-----Halaman ini sengaja dikosongkan-----

LAMPIRAN B

B.1 *Listing Program Arduino Software*

```
char val = 0;

void setup() {
  // put your setup code here, to run once:
  pinMode (37,OUTPUT);
  pinMode (39,OUTPUT);
  pinMode (43,OUTPUT);
  pinMode (45,OUTPUT);
  pinMode (47,OUTPUT);
  pinMode (49,OUTPUT);
  pinMode (41,OUTPUT);
  pinMode (48,OUTPUT);
  pinMode (46,OUTPUT);
  pinMode (42,OUTPUT);
  pinMode (40,OUTPUT);
  pinMode (38,OUTPUT);
  pinMode (36,OUTPUT);
  pinMode (44,OUTPUT);
  pinMode (34,OUTPUT);
  pinMode (32,OUTPUT);
  pinMode (28,OUTPUT);
  pinMode (26,OUTPUT);
  pinMode (24,OUTPUT);
  pinMode (22,OUTPUT);
  pinMode (30,OUTPUT);
  pinMode (35,OUTPUT);
  pinMode (33,OUTPUT);
  pinMode (29,OUTPUT);
  pinMode (27,OUTPUT);
  pinMode (25,OUTPUT);
  pinMode (23,OUTPUT);
  pinMode (31,OUTPUT);
  Serial.begin(9600);
}

void loop() {
```

```

// put your main code here, to run repeatedly:
if(Serial.available ()>0){
  val = Serial.read();
}
if (val == 'a'){
  digitalWrite(37,HIGH);
  delayMicroseconds(1000);
  digitalWrite(37,LOW);
  Serial.print("a");
}
if (val == 'b'){
  digitalWrite(39,HIGH);
  delayMicroseconds(1000);
  digitalWrite(39,LOW);
  Serial.print("b");
}
if (val == 'c'){
  digitalWrite(43,HIGH);
  delayMicroseconds(1000);
  digitalWrite(43,LOW);
  Serial.print("c");
}
if (val == 'd'){
  digitalWrite(45,HIGH);
  delayMicroseconds(1000);
  digitalWrite(45,LOW);
  Serial.print("d");
}
if (val == 'G'){
  digitalWrite(47,HIGH);
  delayMicroseconds(1000);
  digitalWrite(47,LOW);
  Serial.print("G");
}
if (val == 'f'){
  digitalWrite(49,HIGH);
  delayMicroseconds(1000);
  digitalWrite(49,LOW);
  Serial.print("f");
}

```



```

}
if (val == 'g'){
    digitalWrite(41,HIGH);
    delayMicroseconds(1000);
    digitalWrite(41,LOW);
    Serial.print("g");
}
if (val == 'h'){
    digitalWrite(48,HIGH);
    delayMicroseconds(1000);
    digitalWrite(48,LOW);
    Serial.print("h");
}

if (val == 'i'){
    digitalWrite(46,HIGH);
    delayMicroseconds(1000);
    digitalWrite(46,LOW);
    Serial.print("i");
}
if (val == 'j'){
    digitalWrite(42,HIGH);
    delayMicroseconds(1000);
    digitalWrite(42,LOW);
    Serial.print("j");
}
if (val == 'k'){
    digitalWrite(40,HIGH);
    delayMicroseconds(1000);
    digitalWrite(40,LOW);
    Serial.print("k");
}
if (val == 'l'){
    digitalWrite(38,HIGH);
    delayMicroseconds(1000);
    digitalWrite(38,LOW);
    Serial.print("l");
}
if (val == 'm'){

```

```

digitalWrite(36,HIGH);
delayMicroseconds(1000);
digitalWrite(36,LOW);
Serial.print("m");
}
if (val == 'n'){
digitalWrite(44,HIGH);
delayMicroseconds(1000);
digitalWrite(44,LOW);
Serial.print("n");
}
if (val == 'o'){
digitalWrite(34,HIGH);
delayMicroseconds(1000);
digitalWrite(34,LOW);
Serial.print("o");
}
if (val == 'p'){
digitalWrite(32,HIGH);
delayMicroseconds(1000);
digitalWrite(32,LOW);
Serial.print("p");
}
if (val == 'q'){
digitalWrite(28,HIGH);
delayMicroseconds(1000);
digitalWrite(28,LOW);
Serial.print("q");
}
if (val == 'r'){
digitalWrite(26,HIGH);
delayMicroseconds(1000);
digitalWrite(26,LOW);
Serial.print("r");
}
if (val == 's'){
digitalWrite(24,HIGH);
delayMicroseconds(1000);
digitalWrite(24,LOW);

```

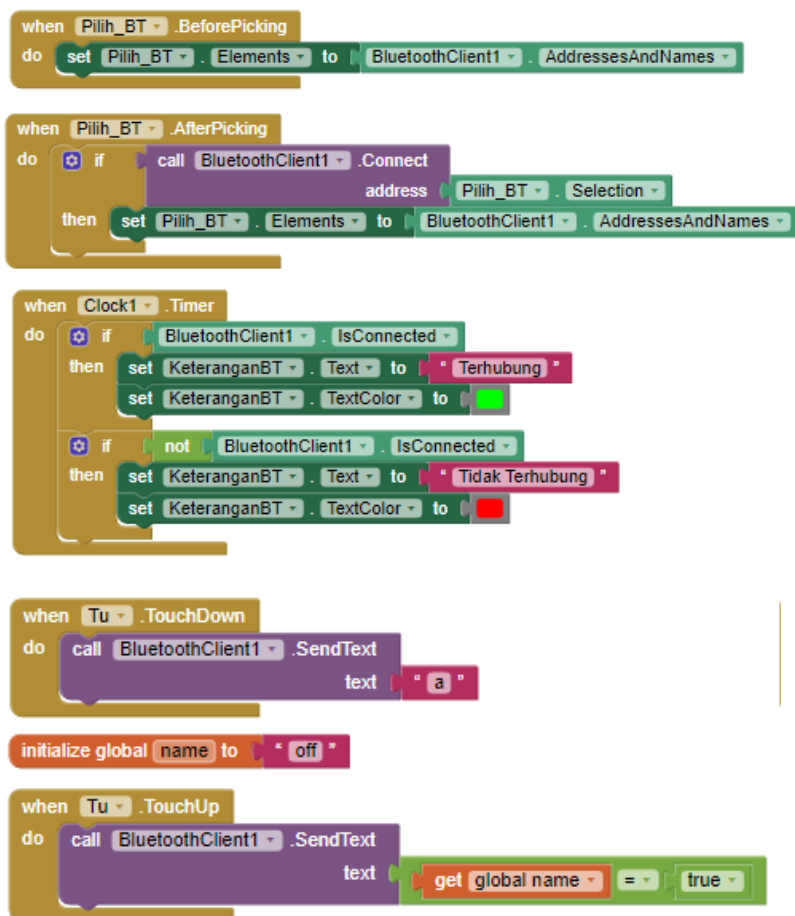
```

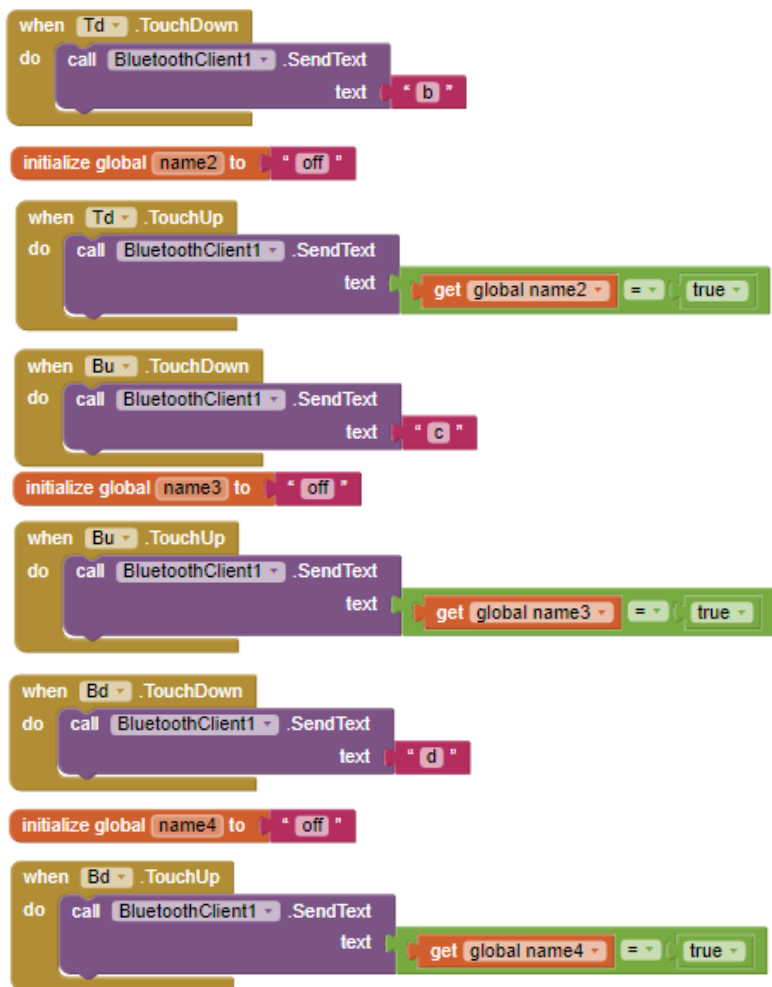
    Serial.print("s");
}
if (val == 't'){
    digitalWrite(22,HIGH);
    delayMicroseconds(1000);
    digitalWrite(22,LOW);
    Serial.print("t");
}
if (val == 'u'){
    digitalWrite(30,HIGH);
    delayMicroseconds(1000);
    digitalWrite(30,LOW);
    Serial.print("u");
}
if (val == 'v'){
    digitalWrite(35,HIGH);
    delayMicroseconds(1000);
    digitalWrite(35,LOW);
    Serial.print("v");
}
if (val == 'w'){
    digitalWrite(33,HIGH);
    delayMicroseconds(1000);
    digitalWrite(33,LOW);
    Serial.print("w");
}
if (val == 'x'){
    digitalWrite(29,HIGH);
    delayMicroseconds(1000);
    digitalWrite(29,LOW);
    Serial.print("x");
}
if (val == 'y'){
    digitalWrite(27,HIGH);
    delayMicroseconds(1000);
    digitalWrite(27,LOW);
    Serial.print("y");
}
if (val == 'z'){

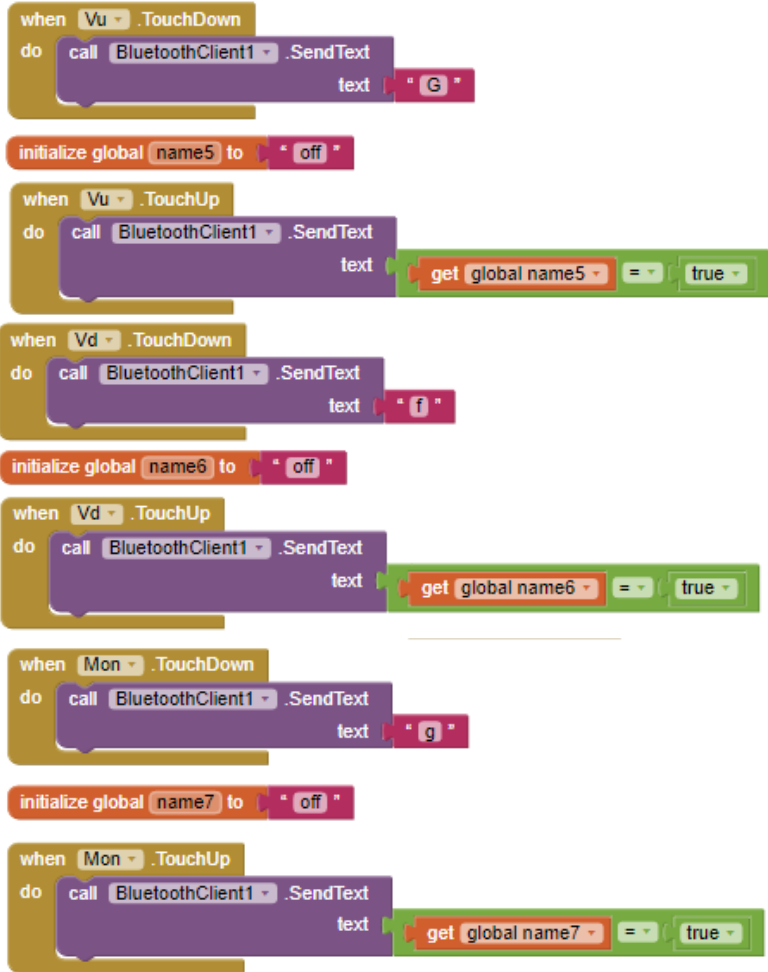
```

```
digitalWrite(25,HIGH);  
delayMicroseconds(1000);  
digitalWrite(25,LOW);  
Serial.print("z");  
}  
if (val == 'A'){  
    digitalWrite(23,HIGH);  
    delayMicroseconds(1000);  
    digitalWrite(23,LOW);  
    Serial.print("A");  
}  
if (val == 'B'){  
    digitalWrite(31,HIGH);  
    delayMicroseconds(1000);  
    digitalWrite(31,LOW);  
    Serial.print("B");  
}  
}
```

B.2 Listing Program MIT App Inventor 2

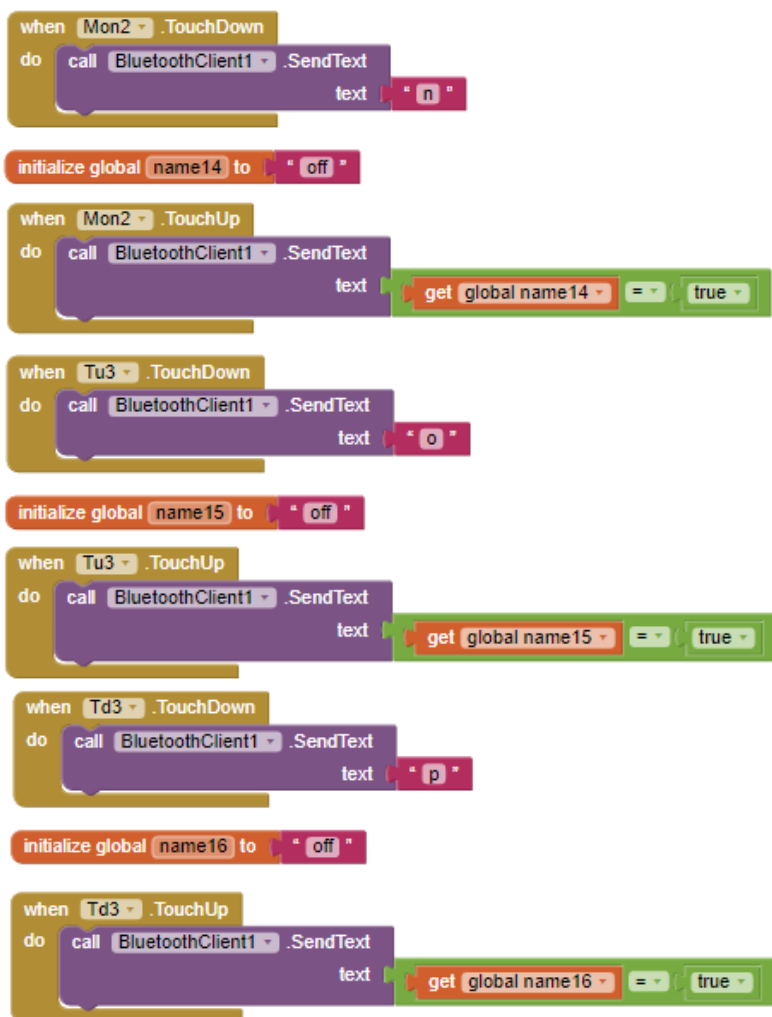




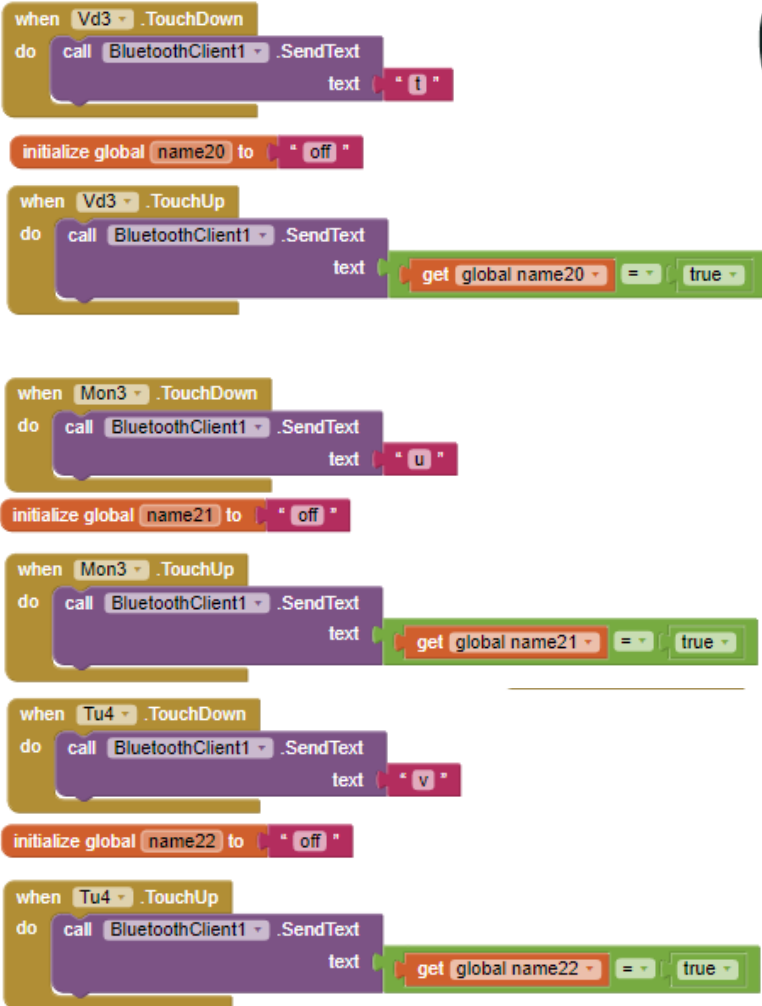






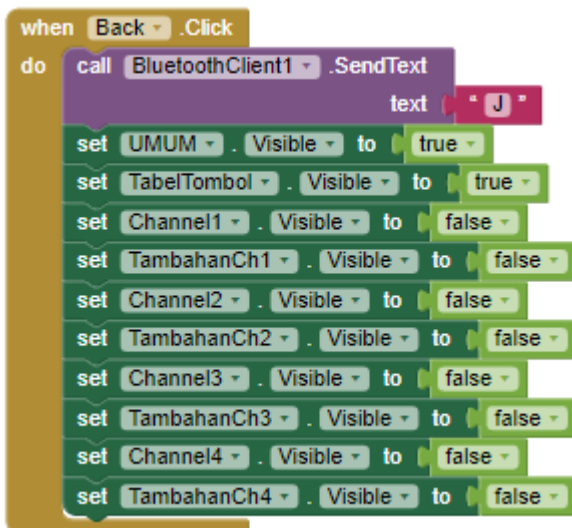
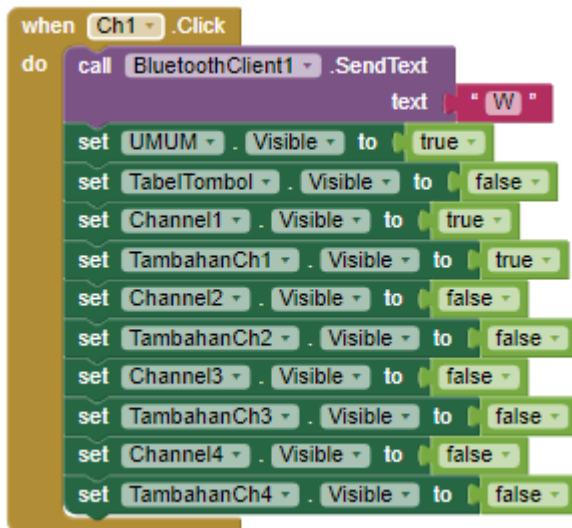


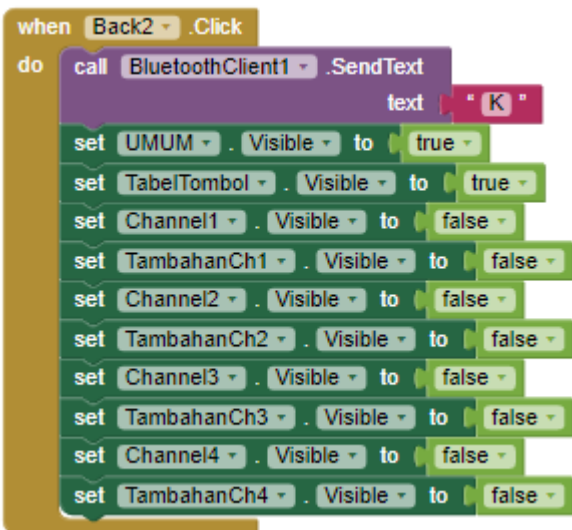
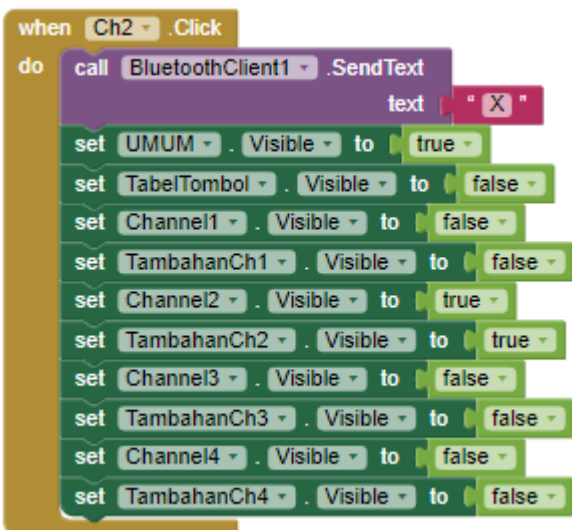


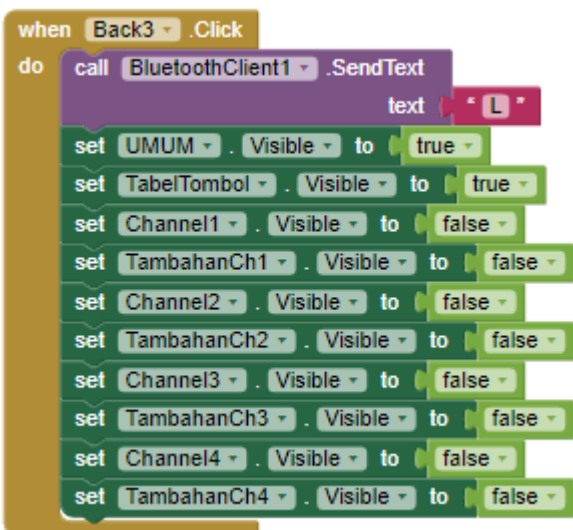
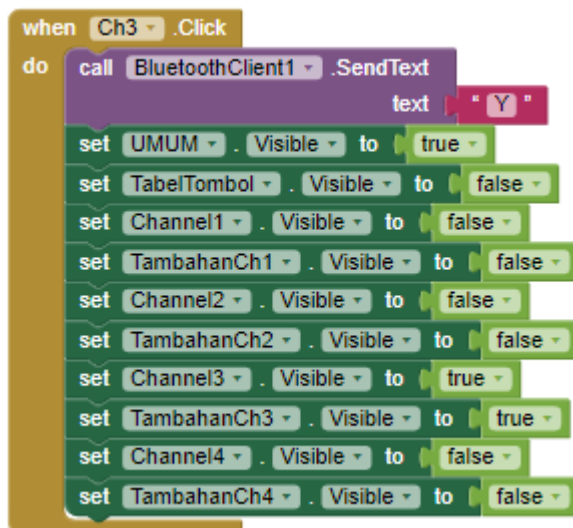












```
when Ch4 .Click
do
  call BluetoothClient1 .SendText
  text " Z "
  set UMUM . Visible to true
  set TabelTombol . Visible to false
  set Channel1 . Visible to false
  set TambahanCh1 . Visible to false
  set Channel2 . Visible to false
  set TambahanCh2 . Visible to false
  set Channel3 . Visible to false
  set TambahanCh3 . Visible to false
  set Channel4 . Visible to true
  set TambahanCh4 . Visible to true
```

```
when Back4 .Click
do
  call BluetoothClient1 .SendText
  text " M "
  set UMUM . Visible to true
  set TabelTombol . Visible to true
  set Channel1 . Visible to false
  set TambahanCh1 . Visible to false
  set Channel2 . Visible to false
  set TambahanCh2 . Visible to false
  set Channel3 . Visible to false
  set TambahanCh3 . Visible to false
  set Channel4 . Visible to false
  set TambahanCh4 . Visible to false
```

LAMPIRAN C

C.1 Data Sheet MAXIM 5406

12-3817; Rev 0; 5/06



Audio Processor with Pushbutton Interface

General Description

The MAX5406 stereo audio processor provides a complete audio solution with volume, balance, bass, and treble controls. It features dual 32-tap logarithmic potentiometers for volume control, dual potentiometers for balance control, and linear digital potentiometers for tone control. A simple debounced pushbutton interface controls all functions. The MAX5406 advances the wiper setting once per button push. Maxim's proprietary SmartWiper™ control eliminates the need for a microcontroller (μ C) to increase the wiper transition rate. Holding the control input low for more than 1s advances the wiper at a rate of 4Hz for 4s and 16Hz thereafter. An integrated click/pop suppression feature eliminates the audible noise generated by the wiper's movements.

The MAX5406 provides a subwoofer output that internally combines the left and right channels. An external filter capacitor allows for a customized cut-off frequency for the subwoofer output. A bass-boost mode enhances the low-frequency response of the left and right channels. An integrated bias amplifier generates the required $(V_{DD} + V_{SS}) / 2$ bias voltage, eliminating the need for external op amps for unipolar operation.

The MAX5406 also features ambience control to enhance the separation of the left- and right-channel outputs for headphones and desktop speakers systems, and a pseudostereo feature that approximates stereo sound from a monophonic signal.

The MAX5406 is available in a 7mm x 7mm, 48-pin TQFN package and in a 48-pin TSSOP package and is specified over the extended $(-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$) temperature range.

Applications

Automotive Rear-Seat Entertainment (RSE)
Desktop Speakers
Portable Audio
PDAs or MP3 Player Docking Stations
Karaoke Machines
Flat-Screen TVs

Features

- ♦ Audio Processor Including All Op Amps and Pots for Volume, Balance, Mute, Bass, Treble, Ambience, Pseudostereo, and Subwoofer
- ♦ 32-Tap Volume Control (2dB Steps)
- ♦ Small, 7mm x 7mm, 48-Pin TQFN and 48-Pin TSSOP Packages
- ♦ Single $+2.7\text{V}$ to $+5.5\text{V}$ or Dual $\pm 2.7\text{V}$ Supply Operation
- ♦ Clickless Switching and Control
- ♦ Mute Function to $< -90\text{dB}$ (typ)
- ♦ Channel Isolation $> -70\text{dB}$ (typ)
- ♦ Two Sets of Single-Ended or Differential Stereo Inputs Can Be Used for Summing/Mixing
- ♦ Debounced Pushbutton Interface Works with Momentary Contact Switches or Microprocessors (μ Ps)
- ♦ Low $0.2\mu\text{A}$ (typ) Shutdown Supply Current
- ♦ Shutdown Stores All Control Settings
- ♦ 0.02% (typ) THD into $10\text{k}\Omega$ Load, $25\mu\text{VRMS}$ (typ) Output Noise
- ♦ Internally Generated $1/2$ Full-Scale Bias Voltage for Single-Ended Applications
- ♦ Power-On Volume Setting to -20dB
- ♦ Internal Passive RF Filters for Analog Inputs Prevent High Frequencies from Reaching the Speakers

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE	PKG CODE
MAX5406EUM	-40°C to $+85^{\circ}\text{C}$	48 TSSOP	U48-1
MAX5406ETM*	-40°C to $+85^{\circ}\text{C}$	48 TQFN	T4877-6

*Future product—contact factory for availability.

Pin Configurations appear at end of data sheet.

SmartWiper is a trademark of Maxim Integrated Products, Inc.



Maxim Integrated Products 1

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

Audio Processor with Pushbutton Interface

ABSOLUTE MAXIMUM RATINGS

L1_H, L1_L, L2_H, L2_L	
to VSS-0.3V to the lower of (VDD + 0.3V) or +6V
R1_H, R1_L, R2_H, R2_L	
to VSS-0.3V to the lower of (VDD + 0.3V) or +6V
AMB, BALL, BALR, VOLUP, VOLDN, MUTE, SHDN, BASSDN, BASSUP, TREBDN, TREBUP	
to DGND-0.3V to the lower of (VLOGIC + 0.3V) or +6V
CTL, CTR, CBL, CBR, CLS, CRS, CSUB, CBIAS, CMSNS, AMBLI, AMBRI, BIAS	
to VSS-0.3V to the lower of (VDD + 0.3V) or +6V
LOUT, ROUT, SUBOUT, LMR, LPR to VSS-0.3V to the lower of (VDD + 0.3V) or +6V

VDD to VSS-0.3V to +6V
VDD to VLOGIC±6V
VLOGIC to DGND-0.3V to +6V
DGND to VSS-0.3V to +6V
LOUT, ROUT, SUBOUT Short Circuited to VSSContinuous
Continuous Power Dissipation (TA = +70°C)	
48-Pin TQFN (derate 27.8mW/°C above +70°C)2222mW
48-Pin TSSOP (derate 16mW/°C above +70°C)1282mW
Operating Temperature Range-40°C to +85°C
Junction Temperature+150°C
Storage Temperature Range-60°C to +150°C
Lead Temperature (soldering, 10s)+300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(VDD = VLOGIC = +5.0V, VSS = 0, VBIAS = VCMSNS = VDD / 2, DGND = 0, ambience disabled, VAMBLI = VAMBRI = VBIAS, VR1_L = VL1_L = VR2_L = VL2_L = external VBIAS, CCSUB = 0.15µF, CCLS = CCRS = 1µF, CQBL = CQBR = 3.3nF, CCTL = CCTR = 4.7nF, CBIAS = 0.1µF, CBIAS = 50µF (see the Typical Application Circuit), TA = TMIN to TMAX unless otherwise specified. Typical values are at TA = +25°C). (Note 1)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Signal-Inputs Input Resistance	R_{IN}	With respect to V_{BIAS}	R_{INH} R_{INL}	8 16	10 20		k Ω
Signal-Inputs Input Capacitance	C_{IN}	With respect to V_{BIAS}		5			pF
RF Rejection		2MHz to 2.4GHz two-tone test, 2/2.01MHz input to 10kHz out		20			dBc
Differential Input Voltage Range	V_{IN}	$V_{DD} = +5V, V_{SS} = 0, V_{CM} = V_{BIAS}$, gain error $\leq +0.5$ dB	-4		+4	V	
		$V_{DD} = +2.7V, V_{SS} = -2.7V, V_{CM} = V_{BIAS}$, gain error ≤ -0.5 dB	-4.5		+4.5		
Common-Mode Input Voltage Range	V_{CM}	$V_{DD} = +5V, V_{SS} = 0, V_{BIAS} = V_{DD} / 2$, $V_{DIFF} = 100$ mV	$V_{SS} + 0.5V$		$V_{DD} - 0.5V$	V	
		$V_{DD} = +2.7V, V_{SS} = -2.7V, V_{BIAS} = 0$, $V_{DIFF} = 100$ mV					
Bias Voltage	V_{BIAS}	Internally generated ($V_{CMSNS} = V_{SS}$)	$(V_{DD} + V_{SS}) / 2$			V	
Bias-Voltage Input Current		$I_{L_H} = I_{L_L} = V_{BIAS}$, $I_{L_J} = I_{L_J} =$ open, $V_{CMSNS} = V_{DD}$	1			mA	
AUDIO PROCESSING FUNCTIONS							
Maximum Balance Difference		(Note 2)		10	12	14	dB
Minimum Balance Difference		(Note 2)			0		dB
Balance Resolution		(Note 2)			2		dB
Maximum Volume Attenuation		(Note 2)		-63	-62	-59	dB
Minimum Volume Attenuation		(Note 2)		-0.5	0	+0.5	dB
Volume Resolution		(Note 2)			2		dB
Volume-Control Steps		(Note 2)			32		steps

Audio Processor with Pushbutton Interface

ELECTRICAL CHARACTERISTICS (continued)

(VDD = VLOGIC = +5.0V, VSS = 0, VBIAS = VCMSENS = VDD / 2, DGND = 0, ambience disabled, VAMBLJ = VAMBRJ = VBIAS, VR1_L = VL1_L = VR2_L = VL2_L = external VBIAS, CCBUS = 0.15μF, CCLS = CCRS = 1μF, CCB_L = CCB_R = 3.3nF, CCT_L = CCT_R = 4.7nF, CBIAS = 0.1μF, CCBIAS = 50μF (see the Typical Application Circuit), TA = TMIN to TMAX unless otherwise specified. Typical values are at TA = +25°C). (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Gain Matching of Input 1 to Input 2 of Each Channel		Volume = 0dB (Note 2)	-0.1		+0.1	dB
Gain Matching of Left to Right Channel		Volume = 0dB (Note 2)	-0.1		+0.1	dB
Bass-Boost Range		fBASS = 1kHz, treble = 0dB, CCB_ = open, CCT_ = open (Note 3)	10	14		dB
Bass-Cut Range		fBASS = 1kHz, treble = 0dB, CCB_ = open, CCT_ = open (Note 3)	10	14		dB
Treble-Boost Range		fTREBLE = 1kHz, bass = 0dB, CCB_ = open, CCT_ = short (Note 3)	10	15		dB
Treble-Cut Range		fTREBLE = 1kHz, bass = 0dB, CCB_ = open, CCT_ = short (Note 3)	10	15		dB
Bass-Boost/-Cut Steps		Max boost to max cut		21		steps
Treble-Boost/-Cut Steps		Max boost to max cut		21		steps
Bass End-to-End Resistance	REPOT			116		kΩ
Treble End-to-End Resistance	RTPT			17		kΩ
Bass Series Resistance	RB			40		kΩ
Treble Series Resistance	RT			3.5		kΩ
Mute Attenuation				-90		dB
AC PERFORMANCE (VIN = 1Vp-p, RL = 10kΩ, VDD = +2.7V, VSS = -2.7V, volume = 0dB, treble = bass = 0dB)						
Total Harmonic Distortion Plus Noise	THD+N	(Notes 4, 5)		0.02	0.06	%
Interchannel Crosstalk		L to R or R to L		-70		dB
ROUT/LOUT OUTPUTS						
Maximum Load Capacitance	CLoad			100		pF
Output-Voltage Swing	VOUT-P	RL = 10kΩ, VDD = +2.7V, VSS = -2.7V	-2.3		+2.3	V
Output Offset Voltage	VODS	VDD = +2.7V, VSS = -2.7V, volume = 0dB, RL = 10kΩ, inputs = VBIAS	-30	0	+30	mV
Short-Circuit Output Current	ISC	Shorted to VSS		15		mA
Output Resistance	R_OUT	ILoad = 100μA to 500μA		10		Ω

Audio Processor with Pushbutton Interface

ELECTRICAL CHARACTERISTICS (continued)

(VDD = VLOGIC = +5.0V, VSS = 0, VBIAS = VCMSENS = VDD / 2, DGND = 0, ambience disabled, VAMBLI = VAMBRI = VBIAS, VR1_L = VL1_L = VR2_L = VL2_L = external VBIAS, CCSUB = 0.15μF, CCLS = CCRS = 1μF, CCLL = CCLR = 3.3nF, CCTL = CCTR = 4.7nF, CBAS = 0.1μF, CBAS = 50μF (see the Typical Application Circuit), TA = TMIN to TMAX unless otherwise specified. Typical values are at TA = +25°C). (Note1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Noise	e_n	$f_{BW} = 20\text{Hz to } 20\text{kHz}$, $V_{IN} = V_{BIAS}$, mute on, noise measured at LOUT and ROUT (Notes 2, 4, 5)		3.5	9.5	μV_{RMS}
		$f_{BW} = 20\text{Hz to } 20\text{kHz}$, $V_{IN} = V_{BIAS}$, mute off, volume = 0dB, noise measured at LOUT and ROUT (Notes 2, 4, 5)		25	35	
Power-Supply Rejection Ratio	PSRR	100mV _{p-p} at 217Hz on V _{DD}		-70		dB
		100mV _{p-p} at 1kHz on V _{DD}		-65		
SUBWOOFER OUTPUT						
Gain		(V _{L1_H} - V _{L1_L}) to (V _{SUBOUT} - V _{BIAS}), volume = 0dB (Note 2)		-6		dB
Highpass Filter Cutoff Frequency		Volume = 0dB		15		Hz
Internal Highpass Cutoff Resistance	R _S	Figure 12		13.8		k Ω
Lowpass Filter Cutoff Frequency		Volume = 0dB		100		Hz
Internal Lowpass Cutoff Resistance	R _{SUB}	Figure 12		10.6		k Ω
Maximum Load Capacitance	C _{SUBLOAD}			100		pF
Output-Voltage Swing	V _{SUBOUT-P}	R _L = 10k Ω , V _{DD} = +2.7V, V _{SS} = -2.7V	-2.3		+2.3	V
Output Offset Voltage	V _{SUBOOS}	V _{DD} = +2.7V, V _{SS} = -2.7V, volume = 0dB, R _L = 10k Ω	-15	0	+15	mV
Short-Circuit Output Current	I _{SUBSC}	Shorted to V _{SS}		12		mA
Output Resistance	R _{SUBOUT}	I _{LOAD} = 100 μ A to 500 μ A			10	Ω
Output Noise	e_n	$f_{BW} = 20\text{Hz to } 20\text{kHz}$, $V_{IN} = V_{BIAS}$, mute on, noise measured at SUBOUT (Notes 2, 4, 5)		9	11	μV_{RMS}
		$f_{BW} = 20\text{Hz to } 20\text{kHz}$, $V_{IN} = V_{BIAS}$, volume = 0dB, mute off, noise measured at SUBOUT (Notes 2, 4, 5)		25	35	
Power-Supply Rejection Ratio	PSRR	100mV _{p-p} at 217Hz on V _{DD}		-70		dB
		100mV _{p-p} at 1kHz on V _{DD}		-65		
PUSHBUTTON CONTACT INPUTS (MUTE, AMB, VOLUP, VOLDN, BALL, BALR, BASSUP, BASSDN, TREBUP, TREBDN)						
Internal Pullup Resistor	R _{PU}			50		k Ω
Single-Pulse Input Low Time	t _{LPW}	Figures 2a, 11a, 11b		30		ms
Repetitive Input Pulse Separation Time	t _{HPW}	Figure 2b, 11a, 11b		40		ms
First Autoincrement Point	t _{A1}	Figure 3		1		s
First Autoincrement Rate	f _{A1}	Figure 3		4		Hz
Second Autoincrement Point	t _{A2}	Figure 3		4		s
Second Autoincrement Rate	f _{A2}	Figure 3		16		Hz

Audio Processor with Pushbutton Interface

ELECTRICAL CHARACTERISTICS (continued)

(V_{DD} = V_{LOGIC} = +5.0V, V_{SS} = 0, V_{BIAS} = V_{CMSENS} = V_{DD} / 2, DGND = 0, ambience disabled, V_{AMB} = V_{AMB} = V_{BIAS}, V_{R1_L} = V_{L1_L} = V_{R2_L} = V_{L2_L} = external V_{BIAS}, C_{SUB} = 0.15μF, C_{CLS} = C_{CRS} = 1μF, C_{CL} = C_{CB} = 3.3nF, C_{CTL} = C_{CTR} = 4.7nF, C_{BIAS} = 0.1μF, C_{BIAS} = 50μF (see the Typical Application Circuit), T_A = T_{MIN} to T_{MAX} unless otherwise specified. Typical values are at T_A = +25°C). (Note1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
DIGITAL INPUTS (V_{LOGIC} > 3.6V) (MUTE, AMB, VOLUP, VOLDN, BALL, BALR, BASSUP, BASSDN, TREBUP, TREBON)						
Input-Voltage High	V _{IH}		2.4			V
Input-Voltage Low	V _{IL}				0.8	V
SHDN Input-Voltage High	V _{IHSHDN}		3.4			V
SHDN Input-Voltage Low	V _{ILSHDN}				0.8	V
Input Leakage Current					±5	μA
Input Capacitance				5		pF
DIGITAL INPUTS (V_{LOGIC} ≤ 3.6V) (MUTE, AMB, VOLUP, VOLDN, BALL, BALR, BASSUP, BASSDN, TREBUP, TREBON)						
Input-Voltage High	V _{IH}		2			V
Input-Voltage Low	V _{IL}				0.6	V
SHDN Input-Voltage High	V _{IHSHDN}		2			V
SHDN Input-Voltage Low	V _{ILSHDN}				0.6	V
Input Leakage Current					±5	μA
Input Capacitance				5		pF
TIMING CHARACTERISTICS						
Wiper Settling Time	t _{WS}	Click/pop suppression inactive, Figures 2a, 11a, 11b		45		ms
POWER SUPPLIES (V_{CMSENS} = V_{SS}, internal bias enabled)						
Supply-Voltage Difference	V _{DD} - V _{SS}				+5.5	V
Positive Analog Supply Voltage	V _{DD}		+2.7		+5.5	V
Negative Analog Supply Voltage	V _{SS}		-2.7		0	V
Dual-Supply Positive Supply Voltage	V _{DD}	V _{SS} = -2.7V	0		+2.7	V
Active Positive Supply Current	I _{DD}	No signal, all logic inputs pulled high to V _{LOGIC} or unconnected, SHDN = V _{LOGIC} , R _L = 10kΩ (Note 6)		10	13	mA
Active Negative Supply Current (Note 6)	I _{SS}	No signal, all logic inputs connected to DGND or V _{LOGIC} , V _{DD} = +5V, V _{SS} = 0	-13	-10		mA
		No signal, all logic inputs connected to DGND or V _{LOGIC} , V _{DD} = +2.7V, V _{SS} = -2.7V	-13	-10		
Shutdown Supply Current (Note 6)	I _{SHDN}	No signal, V _{DD} = 5V, V _{SS} = 0, all logic inputs connected to DGND or V _{LOGIC} , SHDN = DGND		0.2		μA
		No signal, V _{DD} = +2.7V, V _{SS} = -2.7V, all logic at DGND or V _{LOGIC} , SHDN = DGND	I _{DD}	0.2		
			I _{SS}	50		

Audio Processor with Pushbutton Interface

ELECTRICAL CHARACTERISTICS (continued)

($V_{DD} = V_{LOGIC} = +5.0V$, $V_{SS} = 0$, $V_{BIAS} = V_{CMSENS} = V_{DD} / 2$, $DGND = 0$, ambience disabled, $V_{AMBLI} = V_{AMBERI} = V_{BIAS}$, $V_{R1_L} = V_{L1_L} = V_{R2_L} = V_{L2_L} =$ external V_{BIAS} , $C_{CSUB} = 0.15\mu F$, $C_{CLS} = C_{CHS} = 1\mu F$, $C_{CBL} = C_{CER} = 3.3nF$, $C_{CTL} = C_{CTR} = 4.7nF$, $C_{BIAS} = 0.1\mu F$, $C_{CBIAS} = 50\mu F$ (see the Typical Application Circuit), $T_A = T_{MIN}$ to T_{MAX} unless otherwise specified. Typical values are at $T_A = +25^\circ C$). (Note1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Power-Up Time	t_{PU}	Power first applied, $\text{_OUT} = -20dB$		1		s
Wake-Up Time	t_{WU}	From shutdown (Note 7)		1		s
Logic Supply Voltage	V_{LOGIC}	$DGND = 0$, $V_{LOGIC} \leq V_{DD}$	+2.7		V_{DD}	V
Logic Active Supply Current	I_{LOGIC}	No signal, one button pressed, remaining logic inputs connected to V_{LOGIC} or unconnected			150	μA
Logic Shutdown Supply Current		No signal, all logic inputs connected to V_{LOGIC} or unconnected, $\overline{SHDN} = DGND$ (Note 6)		0.2	2	μA

Note 1: All devices 100% production tested at $T_A = +85^\circ C$. Limits over the operating temperature range are guaranteed by design.

Note 2: Treble = bass = 0dB. C_{CB_L} = open, C_{CT_L} = short, left input signal = right input signal = +2V.

Note 3: See Tables 3 and 4 and Figure 7. $V_{DD} = +2.7V$, $V_{SS} = -2.7V$.

Note 4: Guaranteed by design.

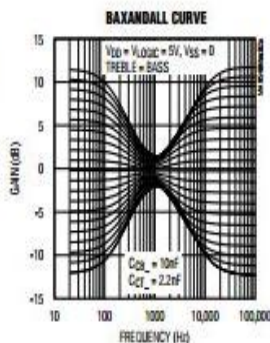
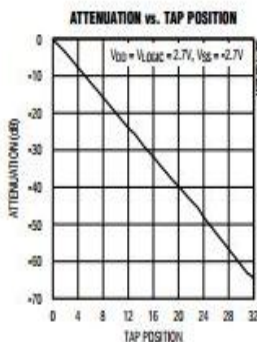
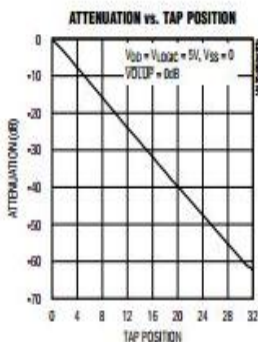
Note 5: Measured with A-weighted filter.

Note 6: Supply current measured while attenuator position is fixed.

Note 7: Set $\text{_OUT} = 0dB$ and shutdown device $\overline{SHDN} = 0$. t_{WU} is the time required for _OUT to reach 0dB after \overline{SHDN} goes high.

Typical Operating Characteristics

($T_A = +25^\circ C$, unless otherwise noted.)



Audio Processor with Pushbutton Interface

Pin Description

PIN		NAME	FUNCTION
TSSOP	TQFN		
1	43	CBIAS	Bypass Capacitor Connection Point to Internally Generated Bias. Bypass CBIAS with a 50 μ F capacitor to system analog ground.
2	44	VSS	Negative Power-Supply Input. Bypass with a 0.1 μ F capacitor to system analog ground.
3	45	L1_H	Left-Channel 1 High Terminal Input. Connect the source between L1_H and L1_L for differential signals. Connect the source to L1_H and tie L1_L to BIAS for single-ended signals.
4	46	L1_L	Left-Channel 1 Low Terminal Input. Connect the source between L1_H and L1_L for differential signals. Connect L1_L to BIAS for single-ended signals.
5	47	L2_L	Left-Channel 2 Low Terminal Input. Connect the source between L2_H and L2_L for differential signals. Connect L2_L to BIAS for single-ended signals.
6	48	L2_H	Left-Channel 2 High Terminal Input. Connect the source between L2_H and L2_L for differential signals. Connect the source to L2_H and tie L2_L to BIAS for single-ended signals.
7	1	LMR	Left Minus Right Output Signal. LMR output provides a signal that is the difference of left and right input signals. See the <i>Ambience Control</i> section for more details.
8	2	AMBLI	Ambience Left-Channel Input. AMBLI provides the proper ambient effect at LOUT based on the transfer function implemented between LMR and AMBLI. See the <i>Ambience Control</i> section for more details.
9	3	CTL1	Left-Channel Treble Tone Control Capacitor Terminal 1. Connect a capacitor between CTL1 and CTL2 to set the treble cutoff frequency. See the <i>Tone Control</i> section for more details.
10	4	CTL2	Left-Channel Treble Tone Control Capacitor Terminal 2. Connect a capacitor between CTL2 and CTL1 to set the treble cutoff frequency. See the <i>Tone Control</i> section for more details.
11	5	CBL1	Left-Channel Bass Tone Control Capacitor Terminal 1. Connect a capacitor between CBL1 and CBL2 to set the bass cutoff frequency. See the <i>Tone Control</i> section for more details.
12	6	CBL2	Left-Channel Bass Tone Control Capacitor Terminal 2. Connect a capacitor between CBL2 and CBL1 to set the bass cutoff frequency. See the <i>Tone Control</i> section for more details.
13	7	LOUT	Left-Channel Output
14	8	CLSN	Subwoofer Left-Channel Highpass Filter Capacitor Negative Terminal. Connect a capacitor between CLSN and CLSP to set the highpass cutoff frequency at SUBOUT. See the <i>Subwoofer Output</i> section for more details.
15	9	CLSP	Subwoofer Left-Channel Highpass Filter Capacitor Positive Terminal. Connect a capacitor between CLSP and CLSN to set the highpass filter cutoff frequency at SUBOUT. See the <i>Subwoofer Output</i> section for more details.
16	10	SUBOUT	Subwoofer Output. Connect a capacitor from SUBOUT to CSUB to set the lowpass filter cutoff frequency at SUBOUT. See the <i>Subwoofer Output</i> section for more details.
17	11	CSUB	Subwoofer Lowpass Filter Capacitor Terminal. Connect a filter capacitor between CSUB and SUBOUT to set the lowpass filter cutoff frequency. See the <i>Subwoofer Output</i> section for more details.
18, 32	12, 26	I.C.	Internally Connected. Connect to DGND.

Audio Processor with Pushbutton Interface

Pin Description (continued)

PIN		NAME	FUNCTION
TSSOP	TQFN		
19	13	MUTE	Active-Low Mute Control Input. Toggles state between muted and not muted. When in the mute state, all wipers are moved to the low end of the volume potentiometers. The last state is restored when MUTE is toggled again. The power-on state is not muted. MUTE is internally pulled up with 50kΩ to V _{LOGIC} .
20	14	VOLDN	Active-Low Downward Volume Control Input. Press VOLDN to decrease the volume. This simultaneously moves left and right volume wipers towards higher attenuation. VOLDN is internally pulled up with 50kΩ to V _{LOGIC} .
21	15	VOLUP	Active-Low Upward Volume Control Input. Press VOLUP to increase the volume. This simultaneously moves the left and right volume wipers towards the lower attenuation. VOLUP is internally pulled up with 50kΩ to V _{LOGIC} .
22	16	BALL	Active-Low Left Balance Control Input. Press BALL to move the balance towards the left channel. BALL is internally pulled up with 50kΩ to V _{LOGIC} .
23	17	BALR	Active-Low Right Balance Control Input. Press BALR to move the balance towards the right channel. BALR is internally pulled up with 50kΩ to V _{LOGIC} .
24	18	DGND	Digital Ground
25	19	V _{LOGIC}	Digital Power-Supply Input. Bypass with 0.1μF to DGND.
26	20	SHDN	Active-Low Shutdown Control Input. In shutdown mode, the MAX5406 stores every wiper's last position. Each wiper moves to the highest attenuation level of its corresponding potentiometer. Terminating shutdown mode restores every wiper to its previous setting. In shutdown, the MAX5406 does not acknowledge any pushbutton command.
27	21	BASSDN	Active-Low Downward Bass Control Input. Press BASSDN to decrease bass boost. Bass boost emphasizes the signal's low-frequency components. BASSDN is internally pulled up with 50kΩ to V _{LOGIC} . To implement a bass-boost button, connect BASSDN to BASSUP. Presses then toggle the state between flat and full bass boost on each button press.
28	22	BASSUP	Active-Low Upward Bass Control Input. Press BASSUP to increase bass boost. Bass boost emphasizes the signal's low frequency components. BASSUP is internally pulled up with 50kΩ to V _{LOGIC} . To implement a bass-boost button, connect BASSUP to BASSDN. Presses then toggle the state between flat and full bass boost on each button press.
29	23	TREBDN	Active-Low Downward Treble Control Input. Press TREBDN to decrease the treble boost. Treble boost emphasizes the signal's high-frequency components. TREBDN is internally pulled up with 50kΩ to V _{LOGIC} .
30	24	TREBUP	Active-Low Upward Treble Control Input. Press TREBUP to increase the treble boost. Treble boost emphasizes the signal's high-frequency components. TREBUP is internally pulled up with 50kΩ to V _{LOGIC} .
31	25	AMB	Active-Low Ambience Switch Control Input. Drive AMB low to toggle on/off the ambience function. AMB is internally pulled up with 50kΩ to V _{LOGIC} .
33	27	CRSN	Subwoofer Right-Channel Highpass Filter Capacitor Negative Terminal. Connect a capacitor between CRSN and CRSP to set the highpass cutoff frequency at SUBOUT. See the Subwoofer Output section for more details.
34	28	CRSP	Subwoofer Right-Channel Highpass Filter Capacitor Positive Terminal. Connect a capacitor between CRSP and CRSN to set the highpass cutoff frequency at SUBOUT. See the Subwoofer Output section for more details.

Audio Processor with Pushbutton Interface

Pin Description (continued)

PIN		NAME	FUNCTION
TSSOP	TQFN		
36	30	CBR2	Right-Channel Bass Tone Control Capacitor Terminal 2. Connect a nonpolarized capacitor between CBR2 and CBR1 to set the bass cutoff frequency. See the <i>Tone Control</i> section for more details.
37	31	CBR1	Right-Channel Bass Tone Control Capacitor Terminal 1. Connect a capacitor between CBR1 and CBR2 to set the bass cutoff frequency. See the <i>Tone Control</i> section for more detail.
38	32	CTR2	Right-Channel Treble Tone Control Capacitor Terminal 2. Connect a capacitor between CTR2 and CTR1 to set the treble cutoff frequency. See the <i>Tone Control</i> section for more details.
39	33	CTR1	Right-Channel Treble Tone Control Capacitor Terminal 1. Connect a capacitor between CTR1 and CTR2 to set the treble cutoff frequency. See the <i>Tone Control</i> section for more details.
40	34	AMBRI	Ambience Right-Channel Input. AMBRI provides the proper ambient effect at ROUT based on the gain between LPR and AMBRI. See the <i>Ambience Control</i> section for more details.
41	35	LPR	Left Plus Right Output Signal. LPR output provides a signal that is a combination of the left and right input signals. See the <i>Ambience Control</i> section for more details.
42	36	VDD	Positive Analog Supply Voltage. Bypass with a 0.1µF capacitor to system analog ground.
43	37	R2_H	Right-Channel High Terminal 2. Connect the source between R2_H and R2_L for differential signal. Connect the source to R2_H and tie R2_L to BIAS for single-ended signals.
44	38	R2_L	Right-Channel Low Terminal 2. Connect the source between R2_H and R2_L for differential signal. Connect R2_L to BIAS for single-ended signals.
45	39	R1_L	Right-Channel Low Terminal 1. Connect the source between R1_H and R1_L for differential signal. Connect R1_L to BIAS for single-ended signals.
46	40	R1_H	Right-Channel High Terminal 1. Connect the source between R1_H and R1_L for differential signal. Connect the source to R1_H and tie R1_L to BIAS for single-ended signals.
47	41	CMSNS	Common-Mode Voltage Sense. Connect to VDD to disable the internal bias generator and drive BIAS with external source to set output DC level.
48	42	BIAS	Internally Generated Bias Voltage. Connect CMSNS to VSS to enable the internally generated $V_{BIAS} = (V_{DD} + V_{SS}) / 2$. Connect a 0.1µF capacitor between BIAS and system analog ground as close to the device as possible. Do not use BIAS to drive external circuitry.

Audio Processor with Pushbutton Interface

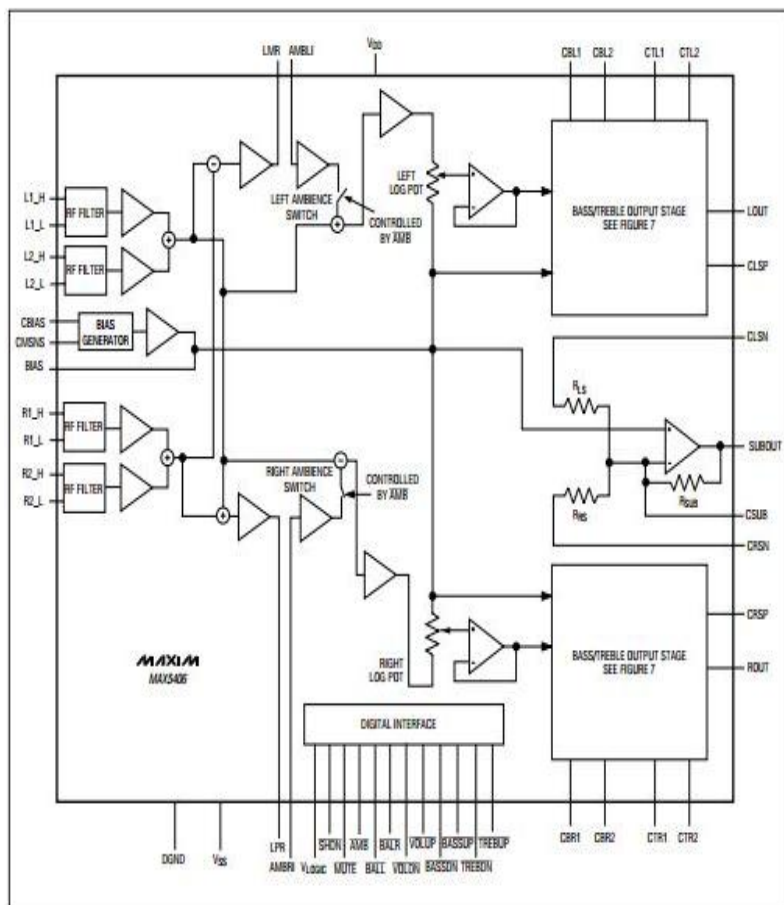
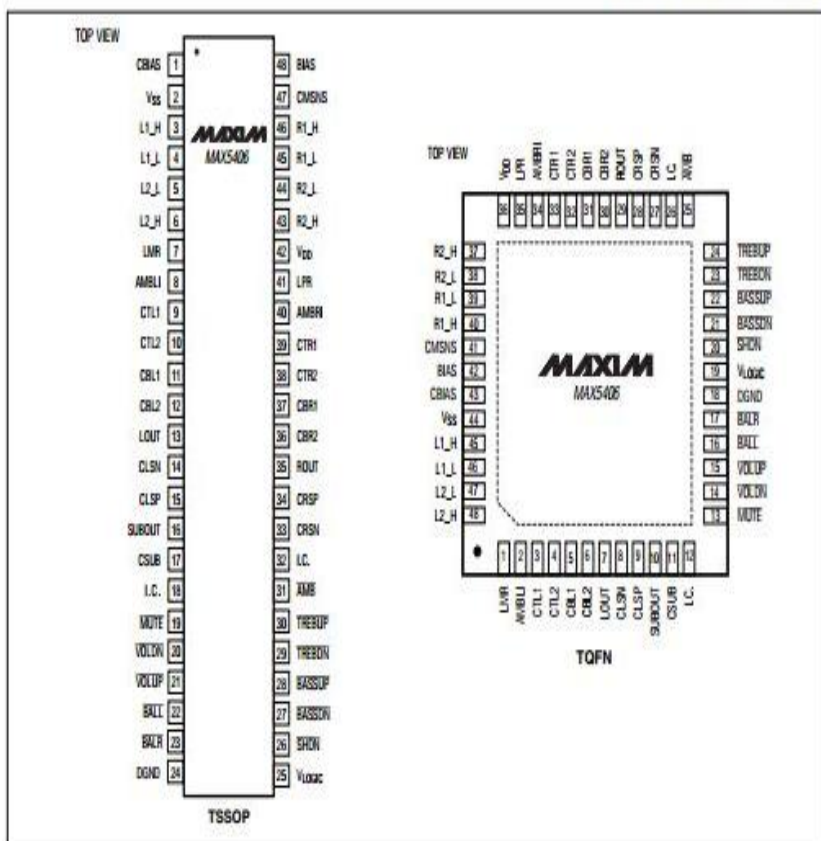


Figure 1. Block Diagram

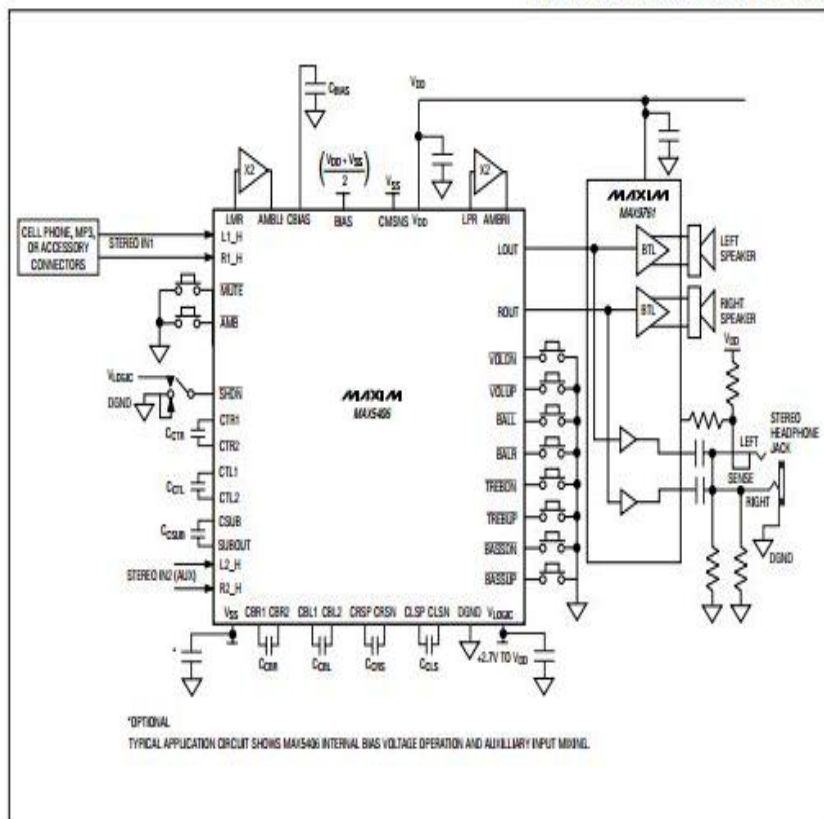
Audio Processor with Pushbutton Interface

Pin Configurations



Audio Processor with Pushbutton Interface

Typical Application Circuit



C.2 Data Sheet Bluetooth HC-05

HC Serial Bluetooth Products User Instructional Manual

1 Introduction

HC serial Bluetooth products consist of Bluetooth serial interface module and Bluetooth adapter, such as:

(1) Bluetooth serial interface module:

Industrial level: HC-03, HC-04(HC-04-M, HC-04-S)

Civil level: HC-05, HC-06(HC-06-M, HC-06-S)

HC-05-D, HC-06-D (with baseboard, for test and evaluation)

(2) Bluetooth adapter:

HC-M4

HC-M6

This document mainly introduces Bluetooth serial module. Bluetooth serial module is used for converting serial port to Bluetooth. These modules have two modes: master and slaver device. The device named after even number is defined to be master or slaver when out of factory and can't be changed to the other mode. But for the device named after odd number, users can set the work mode (master or slaver) of the device by AT commands.

HC-04 specifically includes:

Master device: HC-04-M, M=master

Slave device: HC-04-S, S=slaver

The default situation of HC-04 is slave mode. If you need master mode, please state it clearly or place an order for HC-04-M directly. The naming rule of HC-06 is same.

When HC-03 and HC-05 are out of factory, one part of parameters are set for activating the device. The work mode is not set, since user can set the mode of HC-03, HC-05 as they want.

The main function of Bluetooth serial module is replacing the serial port line, such as:

1. There are two MCUs want to communicate with each other. One connects to Bluetooth master device while the other one connects to slave device. Their connection can be built once the pair is made. This Bluetooth connection is equivalently liked to a serial port line connection including RXD, TXD

signals. And they can use the Bluetooth serial module to communicate with each other.

2. When MCU has Bluetooth slave module, it can communicate with Bluetooth adapter of computers and smart phones. Then there is a virtual communicable serial port line between MCU and computer or smart phone.

3. The Bluetooth devices in the market mostly are slave devices, such as Bluetooth printer, Bluetooth GPS. So, we can use master module to make pair and communicate with them.

Bluetooth Serial module's operation doesn't need drive, and can communicate with the other Bluetooth device who has the serial. But communication between two Bluetooth modules requires at least two conditions:

- (1) The communication must be between master and slave.
- (2) The password must be correct.

However, the two conditions are not sufficient conditions. There are also some other conditions basing on different device model. Detailed information is provided in the following chapters.

In the following chapters, we will repeatedly refer to Linvor's (Formerly known as Guangzhou HC Information Technology Co., Ltd.) material and photos.

2 Selection of the Module

The Bluetooth serial module named even number is compatible with each other; The slave module is also compatible with each other. In other word, the function of HC-04 and HC-06, HC-03 and HC-05 are mutually compatible with each other. HC-04 and HC-06 are former version that user can't reset the work mode (master or slave). And only a few AT commands and functions can be used, like reset the name of Bluetooth (only the slaver), reset the password, reset the baud rate and check the version number. The command set of HC-03 and HC-05 are more flexible than HC-04 and HC-06's. Generally, the Bluetooth of HC-03/HC-05 is recommended for the user.

Here are the main factory parameters of HC-05 and HC-06. Pay attention to the differences:

HC-05	HC-06
Master and slave mode can be switched	Master and slave mode can't be switched
Bluetooth name: HC-05	Bluetooth name: linvor
Password: 1234	Password: 1234

<p>Master role: have no function to remember the last paired slave device. It can be made paired to any slave device. In other words, just set AT+CMODE=1 when out of factory. If you want HC-05 to remember the last paired slave device address like HC-06, you can set AT+CMODE=0 after paired with the other device. Please refer the command set of HC-05 for the details.</p>	<p>Master role: have paired memory to remember last slave device and only make pair with that device unless KEY (PIN26) is triggered by high level. The default connected PIN26 is low level.</p>
<p>Pairing: The master device can not only make pair with the specified Bluetooth address, like cell-phone, computer adapter, slave device, but also can search and make pair with the slave device automatically.</p> <p>Typical method: On some specific conditions, master device and slave device can make pair with each other automatically. (This is the default method.)</p>	<p>Pairing: Master device search and make pair with the slave device automatically.</p> <p>Typical method: On some specific conditions, master and slave device can make pair with each other automatically.</p>
<p>Multi-device communication: There is only point to point communication for modules, but the adapter can communicate with multi-modules.</p>	<p>Multi-device communication: There is only point to point communication for modules, but the adapter can communicate with multi-modules.</p>
<p>AT Mode 1: After power on, it can enter the AT mode by triggering PIN34 with high level. Then the baud rate for setting AT command is equal to the baud rate in communication, for example: 9600.</p> <p>AT mode 2: First set the PIN34 as high level, or while on powering the module set the PIN34 to be high level, the Baud rate used here is 38400 bps.</p> <p>Notice: All AT commands can be operated only</p>	<p>AT Mode: Before paired, it is at the AT mode. After paired it's at transparent communication.</p>

when the PIN34 is at high level. Only part of the AT commands can be used if PIN34 doesn't keep the high level after entering to the AT mode. Through this kind of designing, set permissions for the module is left to the user's external control circuit, that makes the application of HC-05 is very flexible.	
During the process of communication, the module can enter to AT mode by setting PIN34 to be high level. By releasing PIN34, the module can go back to communication mode in which user can inquire some information dynamically. For example, to inquire the pairing is finished or not.	During the communication mode, the module can't enter to the AT mode.
Default communication baud rate: 9600, 4800-1.3M are settable.	Default communication baud rate: 9600, 1200-1.3M are settable.
KEY: PIN34, for entering to the AT mode.	KEY: PIN26, for master abandons memory.
LED1: PIN31, indicator of Bluetooth mode. Slow flicker (1Hz) represents entering to the AT mode2, while fast flicker(2Hz) represents entering to the AT mode1 or during the communication pairing. Double flicker per second represents pairing is finished, the module is communicable. LED2: PIN32, before pairing is at low level, after the pairing is at high level. The using method of master and slaver's indicator is the same. Notice: The PIN of LED1 and LED2 are connected with LED+.	LED: The flicker frequency of slave device is 102ms. If master device already has the memory of slave device, the flicker frequency during the pairing is 110ms/s. If not, or master has emptied the memory, then the flicker frequency is 750ms/s. After pairing, no matter it's a master or slave device, the LED PIN is at high level. Notice: The LED PIN connects to LED+ PIN.
Consumption: During the pairing, the current is	Consumption: During the pairing, the current is

fluctuant in the range of 30-40mA. The mean current is about 25mA. After paring, no matter processing communication or not, the current is 8mA. There is no sleep mode. This parameter is same for all the Bluetooth modules.	fluctuant in the range of 30-40 m. The mean current is about 25mA. After paring, no matter processing communication or not, the current is 8mA. There is no sleep mode. This parameter is same for all the Bluetooth modules.
Reset: PIN11, active if it's input low level. It can be suspended in using.	Reset: PIN11, active if it's input low level. It can be suspended in using.
Level: Civil	Level: Civil

The table above that includes main parameters of two serial modules is a reference for user selection.

HC-03/HC-05 serial product is recommended.

3. Information of Package

The PIN definitions of HC-03, HC-04, HC-05 and HC-06 are kind of different, but the package size is the same: 28mm * 15mm * 2.35mm.

The following figure 1 is a picture of HC-06 and its main PINs. Figure 2 is a picture of HC-05 and its main PINs. Figure 3 is a comparative picture with one coin. Figure 4 is their package size information. When user designs the circuit, you can visit the website of Guangzhou HC Information Technology Co., Ltd. (www.wavesen.com) to download the package library of profile version.



Figure 1 HC-06

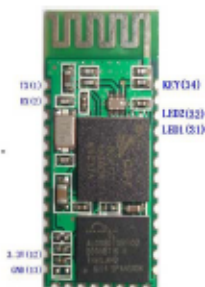


Figure 2 HC-05



Figure 3 Comparative picture with one coin

LINVOX BLUE T
www.linvor.com

LV-BC-2.0

单位: mm

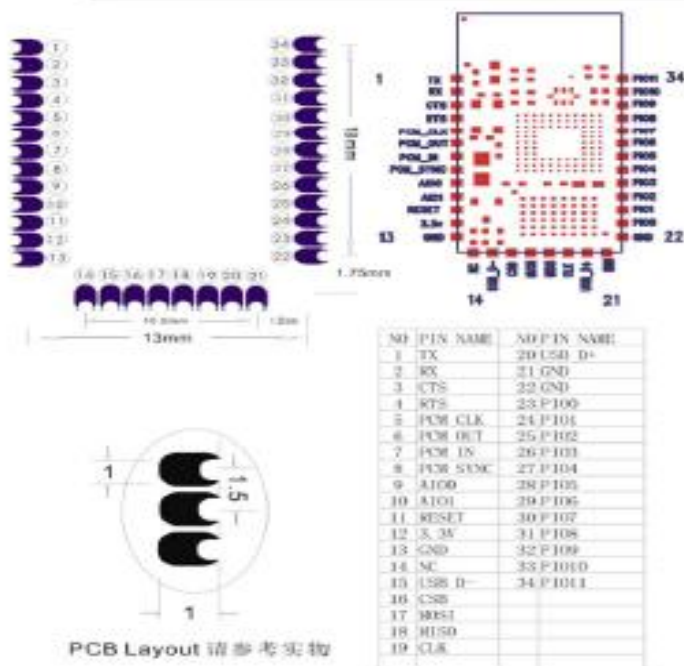


Figure 4 Package size information

4. The Using and Testing Method of HC-06 for the First Time

This chapter will introduce the using method of HC-06 in detail. User can test the module according to this chapter when he or she uses the module at the first time.

PINs description:

PIN1	UART_TXD , TTL/CMOS level, UART Data output
PIN2	UART_RXD, TTL/COMS level, s UART Data input
PIN11	RESET, the reset PIN of module, inputting low level can reset the module, when the module is in using, this PIN can connect to air.
PIN12	VCC, voltage supply for logic, the standard voltage is 3.3V, and can work at 3.0-4.2V
PIN13	GND
PIN22	GND
PIN24	LED, working mode indicator Slave device: Before paired, this PIN outputs the period of 102ms square wave. After paired, this PIN outputs high level. Master device: On the condition of having no memory of pairing with a slave device, this PIN outputs the period of 110ms square wave. On the condition of having the memory of pairing with a slave device, this PIN outputs the period of 750ms square wave. After paired, this PIN outputs high level.
PIN26	For master device, this PIN is used for emptying information about pairing. After emptying, master device will search slaver randomly, then remember the address of the new got slave device. In the next power on, master device will only search this address.

(1) The circuit 1 (connect the module to 3.3V serial port of MCU) is showed by figure 5.

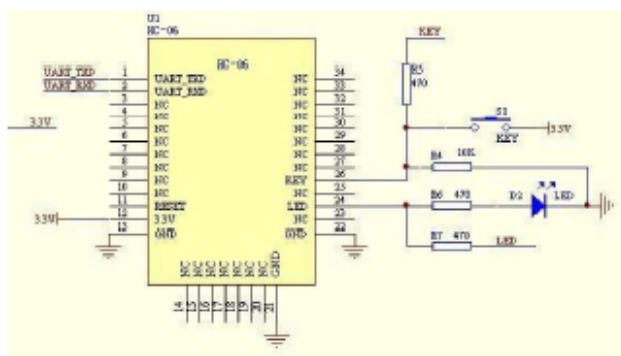


Figure 5 The circuit 1

In principle, HC-06 can work when UART_TXD, UART_RXD, VCC and GND are connected. However, for better testing results, connecting LED and KEY are recommended (when testing the master).

Where, the 3.3V TXD of MCU connects to HC-06's UART_RXD, the 3.3V RXD of MCU connects to HC-06's UART_TXD, and 3.3V power and GND should be connected. Then the minimum system is finished.

Note that, the PIN2:UART_RXD of Bluetooth module has no pull-up resistor. If the MCU TXD doesn't have pull-up function, then user should add a pull-up resistor to the UART_RXD. It may be easy to be ignored.

If there are two MCU which connect to master and slave device respectively, then before paired(LED will flicker) user can send AT commands by serial port when the system is power on. Please refer to HC-04 and HC-06's data sheet for detailed commands. In the last chapter, the command set will be introduced. Please pay attention to that the command of HC-04/HC-06 doesn't have terminator. For example, consider the call command, sending out AT is already enough, need not add the CRLF (carriage return line feed).

If the LED is constant lighting, it indicates the pairing is finished. The two MCUs can communicate with each other by serial port. User can think there is a serial port line between two MCUs.

(2) The circuit 2 (connect the module to 5V serial port of MCU) is showed by figure 6.

Figure 6 is the block diagram of Bluetooth baseboard. This kind of circuit can amplify Bluetooth module's operating voltage to 3.1-5.5V. In this diagram, the J1 port can not only be connected with MCU system of 3.3V and 5V, but also can be connected with computer serial port.

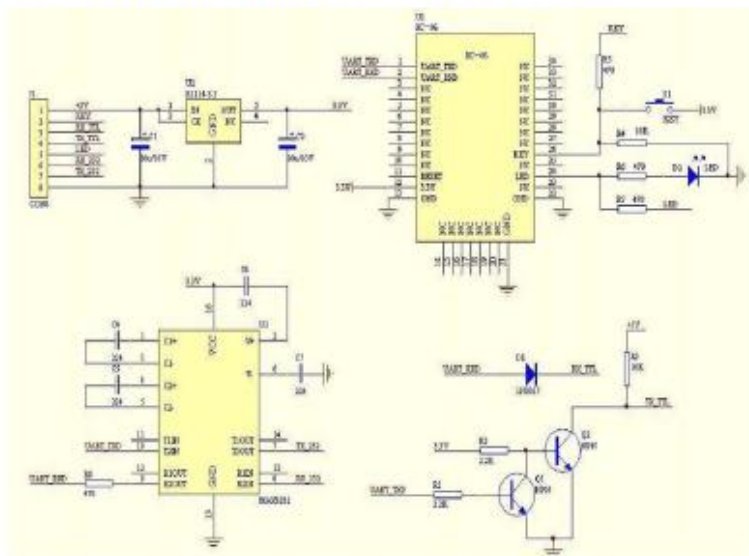


Figure 6 The circuit 2

(3) AT command test

Before paired, the mode of HC-04 and HC-06 are AT mode.

On the condition of 9600N81, OK will be received when user send the two letters AT. Please refer to the last chapter of datasheet for other commands of HC-06. Please pay attention to that sending out AT is already enough, need not add the CRLF (carriage return line feed).

The command set of Version V1.4 doesn't include parity. The version V1.5 and its later version have parity function. Moreover, there are three more commands of V1.5 than V1.4. They are:

No parity (default) AT+PN

(3) Application circuit 2 (connect to 5V serial system or PC serial)

the serial helper is installed, user just need enter "ENTER" key at the end of command.

Reset the master-slave role command:

AT+ROLE=0 ----Set the module to be slave mode. The default mode is slave.

AT+ROLE=1 ----Set the module to be master mode.

Set memory command:

AT+CMODE=1

Set the module to make pair with the other random Bluetooth module (Not specified address). The default is this mode.

AT+CMODE=1

Set the module to make pair with the other Bluetooth module (specified address). If set the module to make pair with random one first, then set the module to make pair with the Bluetooth module has specified address. Then the module will search the last paired module until the module is found.

Reset the password command

AT+PSWD=XXXX

Set the module pair password. The password must be 4-bits.

Reset the baud rate

AT+UART=<Param>,<Param1>,<Param3>.

More information is provided at HC-0305 command set

Example:

AT+UART=9600,0,0 ----set the baud rate to be 9600N81

Reset the Bluetooth name

AT+NAME=XXXXXX

Summary:

HC-05 has many functions and covers all functions of HC-06. The above commands are the most common ones. Besides this, HC-05 leaves lots of space for user. So HC-05 is better than HC-06 and

recommended. HC-03 is similar with HC-05. The above introduction also suits HC-03

The following reference about HC-03 and HC-05 can be downloaded from company website www.wavesen.com:

HC-03 datasheet .pdf	(the command set introduction is included)
HC-05 datasheet .pdf	(the command set introduction is included)
IVT BlueSoleil-2.6	(IVT Bluetooth drive test version)
Bluetooth FAQ .pdf	
PCB package of Bluetooth key modules	(PCB package lib in protel)
IVT software manual.pdf	(introduce how to operate the modern and make pair with Bluetooth module)
PDA serial test helper.exe	(serial helper used for WM system)
HC-03/05 Bluetooth serial command set.pdf	

6. Ordering information

The website of Guangzhou HC Information Technology Co., Ltd is www.wavesen.com. The contact information is provided at the company website.

Order Way: If you want our product, you can give order to the production center of our company directly or order it in Taobao. There is a link to Taobao in our company website.

Package: 50 pieces chips in an anti-static blister package. The weight of a module is about 0.9g. The weight of a package is about 50g.



C.3 Data Sheet Arduino Mega 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical
Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms &
Conditions

Page 7

Environmental Policies
half sqm of green via Impatto Zero®

Page 7

Technical Specification

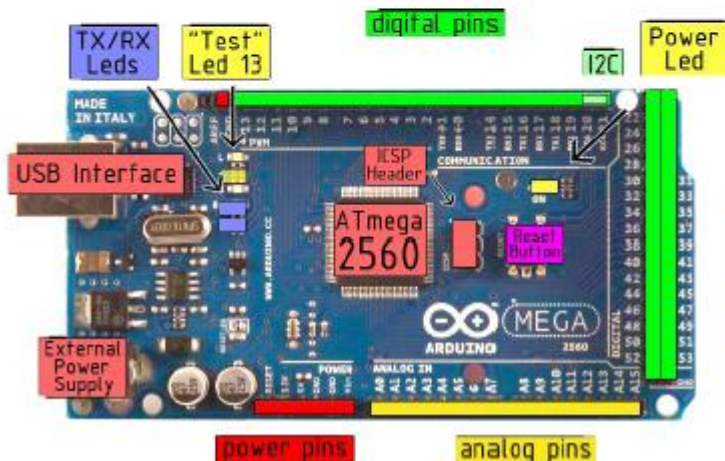


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an Input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 18 (RX) and 19 (TX); Serial 2: 17 (RX) and 18 (TX); Serial 3: 16 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 (Interrupt 0), 3 (Interrupt 1), 18 (Interrupt 6), 19 (Interrupt 4), 20 (Interrupt 3), and 21 (Interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 0 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 50 (MISO), 51 (MOSI), 52 (SS), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C:** 20 (SDA) and 21 (SCL). Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .Inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. Please note that PC is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).

How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

Once you have your skeeth you'll see something very close to the screenshot on the right.

In Tools>Board select MEGA

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.



Done compiling

Press Compile button
(to check for errors)



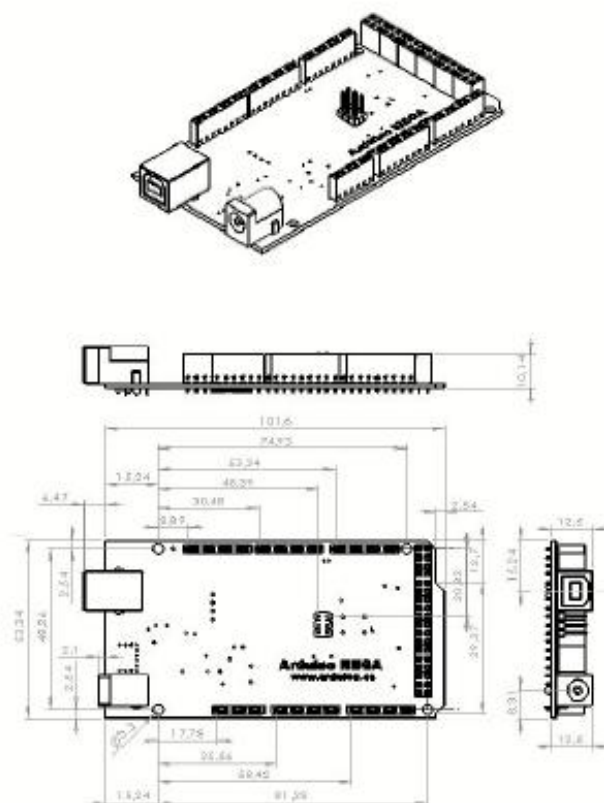
Upload



TX RX Flashing



Blinking Led!



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mismanagement by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any application-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristic of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared km of Costa Rica's forest's.

DAFTAR RIWAYAT HIDUP PENULIS 1



Nama : Dyah Ayuningtyas
TTL : Kediri, 03 Juni 1996
Jenis Kelamin : Perempuan
Agama : Islam
Alamat Rumah : Dsn. Tanggung Ds. Tanggung
002/007 Kec. Campurdarat
Tulungagung
No. HP : 085791208906
Email : ayuningtyashidayah@gmail.com
Hobi : Mendengarkan musik,
menonton film

RIWAYAT PENDIDIKAN

- 2002-2008 : SDN Tanggung III
- 2008-2011 : SMPN 1 Campurdarat
- 2011-2014 : SMAN 1 Boyolangu
- 2014-sekarang : Program Studi Komputer Kontrol Departemen Teknik Elektro Otomasi Fakultas Vokasi ITS

PENGALAMAN KERJA

- PT.TELKOMUNIKASI INDONESIA Tbk
Lab Kalibrasi Telkom MSC Area V Jatim

PENGALAMAN ORGANISASI

- Staff DPM ITS Bagian Eksternal
- Asisten Bendahara HIMAD3TEKTRO ITS 2015/2016
- Bendahara Umum HIMAD3TEKTRO ITS 2016/2017

PENGALAMAN KEPANITIAAN

- Staff Sie Kestari FTI *Olympic Games* (FOG) 2015
- Staff Sie Akomodasi IARC (*Industrial Automatic Robot Competition*) 2015
- Staff Sie Akomodasi IARC (*Industrial Automatic Robot Competition*) 2016
- Staff Sie Acara HTBH (*Health To Be Habbit*) Kesma BEM ITS 2015

-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP PENULIS 2



Nama	: Nafiqul Ihsan
TTL	: Ponorogo, 1 Desember 1995
Jenis Kelamin	: Laki-laki
Agama	: Islam
Alamat	: Jl. Kumbokarno No41 PO
Telp/HP	: 0857 356 11 811
E-mail	: ihsan.nafiqul@yahoo.com

RIWAYAT PENDIDIKAN:

- 2002-2008 : SD Muhammadiyah 1 Ponorogo
- 2008-2011 : SMP Negeri 1 Ponorogo
- 2011-2014 : SMA Negeri 2 Ponorogo
- 2014-2017 : Program Studi Komputer Kontrol Departemen Teknik Elektro Otomasi Fakultas Vokasi ITS

PENGALAMAN KERJA

- Kerja Praktek di PT PLN APD Jawa Timur

PENGALAMAN ORGANISASI

- Kepala Departemen Sosial Masyarakat HIMAD3TEKTRO Transformasi 2016/2017
- Koor Lego NXT Kategori SD IARC (*Industrial Automation and Robotic Competition*) 2016
- Staff Lego NXT Kategori SD IARC (*Industrial Automation and Robotic Competition*) 2015
- Staff Departemen Sosial Masyarakat HIMAD3TEKTRO Solidaritas 2015/2016
- *Basic Electronic Laboratory Assistant* 2016/2017 D3 Teknik Elektro Otomasi

-----Halaman ini sengaja dikosongkan-----